Vehicle Sharing Systems Pricing Optimization

Ariel Waserhole

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques-Informatique**

Arrêté ministérial : 7 août 2006

Présentée par

Ariel WASERHOLE

Thèse dirigée par **Nadia BRAUNER**
et co-encadrée par **Vincent JOST**

préparée au sein du laboratoire **G-SCOP (Grenoble Science pour la Conception et l'Optimisation de la Production)**
et de l'école doctorale **MSTII (Mathématiques, Sciences et Technologies de l'Information, Informatique)**

Optimisation des systèmes de véhicules en libre service par la tarification

(Vehicle Sharing Systems Pricing Optimization)

Thèse soutenue publiquement le **18 novembre 2013**,
devant le jury composé de :

Mme Nadia BRAUNER
Professeur, Université Joseph Fourrier, Grenoble, France, Directeur de thèse

Mr Vincent JOST
CR1 CNRS, Laboratoire G-SCOP, Grenoble, France, Co-Encadrant de thèse

Mr Tal Raviv
Senior lecturer, Tel Aviv University, Israel, Rapporteur

Mr Louis-Martin Rousseau
Professeur agrégé, École Polytechnique de Montréal, Canada, Rapporteur

Mr Frédéric Gardi
Chef de service adjoint Optimisation, Bouygues e-lab, Paris, France, Examinateur

Mr Frédéric Meunier
Chercheur HDR, Ecole Nationale des Ponts et Chaussées, Paris, France, Examinateur

The scientist is not a person who gives the right answers, he's one who asks the right questions.

Claude Lévi-Strauss
(1908–2009)

Short abstract

One-way Vehicle Sharing Systems (VSS), in which users pick-up and return a vehicle in different places is a new type of transportation system that presents many advantages. However, even if advertising promotes an image of flexibility and price accessibility, in reality customers might not find a vehicle at the original station (which may be considered as an infinite price), or worse, a parking spot at destination. Since the first Bike Sharing Systems (BSS), problems of vehicles and parking spots availability have appeared crucial. We define the system performance as the number of trips sold (to be maximized). BSS performance is currently improved by vehicle relocation with trucks. Our scope is to focus on self regulating systems through pricing incentives, avoiding physical station balancing. The question we are investigating in this thesis is the following: Can a management of the incentives increases significantly the performance of the vehicle sharing systems?

Keywords: Vehicle Sharing Systems; Pricing policy; Markov Decision Process (MDP); Scenario-based approach; Fluid approximation; Simulation; Queuing networks; Linear Programming; Approximation algorithm; Complexity.

Résumé court

Nous étudions les systèmes de véhicules en libre service en aller-simple : avec emprunt et restitution dans des lieux éventuellement différents. La publicité promeut l'image de flexibilité et d'accessibilité (tarifaire) de tels systèmes, mais en réalité il arrive qu'il n'y ait pas de véhicule disponible au départ, voire pire, pas de place à l'arrivée. Il est envisageable (et pratiqué pour Vélib' à Paris) de relocaliser les véhicules pour éviter que certaines stations soient vides ou pleines à cause des marées ou de la gravitation. Notre parti-pris est cependant de ne pas considérer de "relocalisation physique" (à base de tournées de camions) en raison du coût, du trafic et de la pollution occasionnées (surtout pour des systèmes de voitures, comme Autolib' à Paris). La question à laquelle nous désirons répondre dans cette thèse est la suivante : Une gestion via des tarifs incitatifs permet-elle d'améliorer significativement les performances des systèmes de véhicules en libre service ?

Mots clés : Véhicules en libre service ; Politiques tarifaires ; Processus de décision markovien ; Approche par scénario ; Approximation fluide ; Simulation ; Réseau de files d'attentes ; Programmation linéaire ; Algorithme d'approximation ; Complexité.

Contents

Remerciements

Tout à commencé à l'ENSIMAG dans l'obscurité d'une salle, à optimiser le déneigement de routes grâce à des *variables*. Avec Grillux et *l'excellent* Jéré, Wojceich Bienia nous avait ferré, la Recherche Opérationnelle allait être notre destinée. Sorti par la filière ingénieur, Amadeus naturellement se présenta, c'est donc près de Nice avec Jordi Planadecursach que je fis mes premiers pas.

Quitter le berceau (Ndlr le Y Grenoblois), me donna des ailes, je fis une première migration Canadienne à Montréal la belle. À Oméga Optimisation, la confection d'horaires d'aéroport me donna envie de voler plus haut. Mon boss courant le professeur Louis-Martin Rousseau visa juste, il me fallait devenir docteur en RO.

Retour à la source, Gre, sa Vence mon eau de jouvence. En quête d'une thèse, d'un Maître, je me remémore ma 2A, mon stage à G-SCOP avec Julien Moncel. C'était décidé je savais où je voulais en être ! Il fallait cependant prendre le temps de s'y mettre. Un petit projet indus' à optimiser des chips avec Yann Kieffer pour ST Micro-électronics. Un approfondissement par Master ROCO. Une optimisation de tournée de VRP avec Pierre Lemaire et Van-Dat Cung. Finalement, dans un train pour une ROADEF, je découvre la bête mathématique, Vincent Jost. Avec Nadia Brauner à trois on s'associe, c'est bon la France me finance ; hip hip hip trois ans de freelance c'est parti !

Trois ans de recherche entre Grenoble, Paris, Lisbon, Clermont-Ferrand, Minsk, Vancouver, Boston, Montréal, Troye, Tenerife et Rome. Trois années au cours desquelles j'ai eu la chance de collaborer avec notamment Yury Orlovich au Belarusian State University, avec Tom McCormick à UBC Ô Canada, ainsi que plus proche avec Jean-Philippe Gayon à Gre. Trois ans (et une brouette) qu'il a fallu rédiger et “tout” rendre en temps constant. Merci à mes rapporteurs, à Louis-Martin Rousseau pour avoir fermé la boucle et à Tal Raviv pour ses commentaires pertinents. Merci à mes examinateurs Frédéric Gardi et Frédéric Meunier pour leur enthousiasme et leurs encouragements.

Je tiens à remercier chaleureusement toutes les personnes citées ci-dessus. Après les faits historisés, place maintenant à ceux qui m'ont suivis dans la continuité. Premièrement j'exprime ma gratitude et mes remerciements les plus sincères à mes directeurs de thèse, Nadia Brauner et Vincent Jost. Leur complémentarité dans l'accompagnement m'a permis de me structurer, de me former et de me perfectionner sur une très vaste palette de sujets autant techniques, philosophiques que stratégiques. Merci Nadia pour tes conseils et discussions emplis d'humanité et de bienveillance. Tu m'as permis de prendre du recul et de reconnecter avec le réel aux moments où il le fallait. J'ai beaucoup apprécié malgré toutes tes responsabilités, que tu aies toujours été là pour moi avec ton rire et ta bonne humeur. Merci Vincent, tu as été mon guide à travers les intempéries de la science moderne. Véritable philosophe, tu m'as entrainé avec fougue et passion sur le fil de tes intuitions, de tes envies, avec comme cap toujours cette exigence de beauté. Notre complicité au cours de ces trois années aura été le fil rouge de cette thèse.

Une épopée de recherche prolifique passe par des co-bureaux aguerris ; merci Julien et Michaël pour le support (me supporter ?), pour vos conseils ainsi que pour votre patience sur mes sempiternelles questions : Comment on dit ? Comment on fait ? C'est quoi ça ! Ou pire : Tu crois que ? Merci G-SCOP pour ta si bonne ambiance. Merci aux interlocuteurs, acteurs ou même simples spectateurs du RU pour ces discussions enflammées. À ce sujet je me dois de citer mon comparse théoricien Bertrand pour sa participation à notre *éveil* sur des textes fondateurs tels que "Pour qu'une théorie soit vraie..." ou encore la *Défense Bérengère Mathieu* ! Merci à l'ADOC et à tous *ses amis* qui ont contribué à ce *team spirit*. Je me remémorerai longtemps nos sessions de *team building* : *e.g.* l'Aventures avec Olivier sur cette fameuse voie mal équipé du Mont Aiguille, ou encore celle avec notre Russe portant notre Espagnole en hypothermie sous la neige / au clair de lune / par -20 degrés / sur 300 mètres de déniv' / pour atteindre finalement un refuge mal chauffé ! Merci aux joueurs de Backgammon, de Wanted, de coinches. Et enfin merci les amis pour *La Chanson* de fin. Je ne pouvais pas rêver mieux que ce "Goodbye Arielou" !

Merci à la Musique et aux musiciens qui m'ont accompagnés durant ces trois années. Bien sûr la base *Marcel Skabouche & les rondelles sanguinolentes* mais également *Aggravated Funking* à Vancouver et enfin *Funky-Peu*.

A thought for my friends from Vancouver, Martin, Nicky, Kurt, Pierre, Claire and Renee. Merci à mes amis de Paris pour m'avoir hébergé tant de fois et fait découvrir la Capital, Gragnes, Jéré, *feu* Bragay et ma Fanny qui m'a si bien accompagné ces derniers mois. Au niveau du..., big up au K-ViocS crew, inébranlable, qui envoie toujours autant le paté ! Ba, Beu, Bru, Bross, Briou, Dart, Dav, Gro, Jean-Bi-Batte, Racho, Pec, Teut, Wag, Zazaï, le clan des Où Marjo, Nathou, Tinou, Maré, Moon, Jah.

Merci Maman, merci Papa pour votre constant soutien. J'apprends tous les jours à vos cotés ; vos "instructions" par le partage de vos passions sont si différentes et en même temps si complémentaires. Je tiens aussi à remercier toute la familia d'être là ! Merci Mum, Peuf, Sarah a.k.a. Rhaled poule-poule, Plicouse, Shakti, Véro, Jeannot, Pec (Laveurl), Pec (Maff), Tom, Fonf et Tirjou. Pour conclure, en écrivant ses lignes j'ai également une pensé pour ceux qui ne sont pas là aujourd'hui mais toujours autant présents dans mon cœur Eliette, Milou et Bob.

Introduction – The thesis

When you can measure what you are talking about and express it in numbers, you know something about it.

Lord William Thomson Kelvin (1824–1907)

In English

Context

Based on a sample of 22 US studies, Shoup (2005) reports that car drivers looking for a parking spot contribute to 30% of the city traffic. Moreover cars are used less than 2 hours per day on average but still occupy a parking spot the rest of the time! Could we have fewer vehicles and satisfy the same demand level?

Recently, the interest in Vehicle Sharing Systems (VSS) in cities has increased significantly. Indeed, urban policies intend to discourage citizens to use their personal car downtown by reducing the number of parking spots, street width, etc. VSS seem to be a promising solution to reduce jointly traffic and parking congestion, noise, and air pollution (proposing bikes or electric cars). They offer personal mobility allowing users to pay only for the usage (sharing the cost of ownership).

We are interested in short-term one-way VSS in which vehicles can be taken and returned at different places. Associated with classic public transportation systems, short-term one-way VSS help to solve one of the most difficult public transit network design problems: the last kilometer issue (DeMaio, 2009). Round-trip VSS, where vehicles have to be returned at the station where they were taken, cannot address this issue.

The first large-scale short-term one-way VSS was the Bicycle Sharing System (BSS) Vélib'. It was implemented in Paris in 2007. Today, it has more than 1200 stations and 20 000 bikes selling around 110 000 trips per day. It has inspired several other cities all around the world: Now more than 300 cities have such a system, including Montréal, Bejing, Barcelona, Mexico City, Tel Aviv (DeMaio, 2009).

One-way Vehicle Sharing Systems: a management issue

One-way systems increase the user freedom at the expense of a higher management complexity. In round trip rental systems, while managing the yield, the only stock that is relevant is the number of available vehicles. In one-way systems, vehicles are not the only key resource anymore: parking stations may have limited number of spots and the available parking spots become an important control leverage.

Since first BSS, problems of bikes and parking spots availability have appeared recursively. Côme (2012), among others, applies data mining to operational BSS data. He offers insights on typical usage patterns to understand causes of imbalances in the distribution of bikes. Reasons are various but we can highlight two important phenomenons: the gravitational effect which indicates that a station is constantly empty or full (as Montmarte hill in Vélib'), and the tide phenomenon representing the oscillation of demand intensity during the day (as morning and evening flows between working and residential areas).

To improve the efficiency of the system, different perspectives are studied in the literature. At a strategic level, some authors consider the optimal capacity and locations of stations; see *e.g.* Shu *et al.* (2010), Lin and Yang (2011) and Kumar and Bierlaire (2012). At a tactical level, other authors investigate the optimal number of vehicles given a set of stations; see *e.g.* George and Xia (2011) and Fricker and Gast (2012). At an operational level, in order to be able to meet the demand with a reasonable standard of quality, in most BSS, trucks are used to balance the bikes among the stations; see *e.g.* Nair and Miller-Hooks (2011), Chemla *et al.* (2012), Contardo *et al.* (2012) and Raviv *et al.* (2013). The objective is to minimize the number of users who cannot be served, counting those who try to take a bike from an empty station or to return it to a full station. The balancing problem amounts to scheduling truck routes to visit stations performing pickup and delivery.

Towards VSS regulated with incentives

A new type of VSS has appeared recently: one-way car VSS with Autolib' in Paris and Car2go in more than 15 cities (Vancouver, San Diego, Amsterdam, Ulm...).

Due to the size of cars, operational balancing optimization through relocation with trucks seems inappropriate. Another way for optimizing the system has to be found.

From an experimental point view, pricing heuristics are studied by Chemla *et al.* (2013) and Pfrommer *et al.* (2013). They appear to perform well in their simulations. However, they do not provide any analytical/mathematical insight on the potential gain of a pricing optimization. Fricker and Gast (2012) consider the optimal sizing of a fleet in "toy" cities, where demand is constant over time and identical for every possible trip, and all stations have the same capacity $\mathcal{K}$. They show that even with an optimal fleet sizing in the most "perfect" city, if there is no operational system management, there is at least a probability of $\frac{2}{\mathcal{K}+1}$ that any given station is empty or full. Fricker and Gast (2012) analyze a heuristic, that can be seen as a dynamic pricing, called "power of two choices": When a user arrives at a station to take a vehicle, he gives randomly two possible destination stations and the system is directing him toward the least loaded one. For their perfect cities, they show that this policy allows to drastically reduce the probability to be empty or full for each station from $\frac{2}{\mathcal{K}+1}$ to $2^{-\frac{\mathcal{K}}{2}}$.

This thesis is investigating pricing optimization for self regulation in VSS. Using operations research we want to estimate the potential impact of using pricing techniques to influence user choices in order to drive the system towards its most efficient dynamic.

Thesis overview – Main contributions

The objective of this thesis is to study the interest of pricing strategies for Vehicle Sharing Systems (VSS) optimization. Figure 1 summarizes our approach giving an overview of the different chapters. We now briefly detail the contributions of each chapter and their dependencies.

In this introduction, we raise an informal question: can we drive users toward the system's most profitable direction by playing on the price to take a trip? An intuitive idea of such pricing policy is given in Figure 1 for a VSS composed with 3 stations and 8 vehicles. This policy is based on the *resources* availability. For instance, the resource vehicle is highly available at the campus station, and the resource parking spot is also highly available at the railway station. Therefore, taking a trip from the campus to the railway station is cheap (\$). Yet, taking the trip on the opposite direction is expensive (\$\$\$), because the resource vehicle at the railway station and the resource parking spot at the campus station are precious. Notice also that, in this example, stations have finite capacities (4, 4 and 3) and that there is a reservation of parking spot at destination (crosses).

In Chapter 1, we give a general overview of the VSS optimization. We present

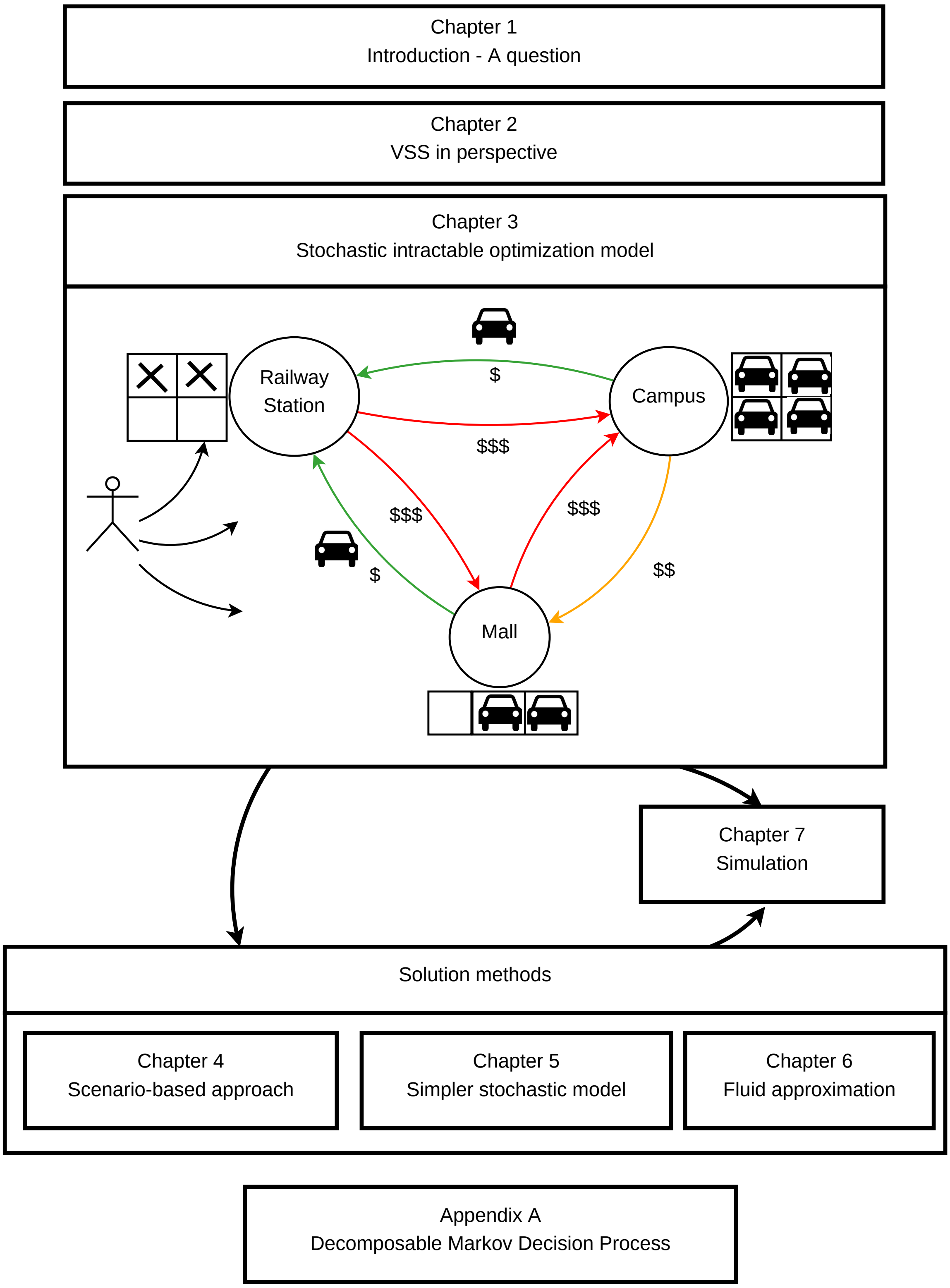


Figure 1: Plan of the thesis, dependency graph of the chapters: $a \to b$ read a before b.

different facets of VSS management and the leverages available to optimize it. We detail a terminology for the pricing optimization in VSS that allow us to present a classified literature review.

In Chapter 2, we propose a formal stochastic model that allows us to define the VSS stochastic pricing problem. This problem is our reference, the "Holy Grail" that we try to solve all along this thesis. This chapter can be read independently, although Chapter 1 allows to put in context the chosen assumptions and simplifications. Even if this stochastic model is simple, because of the granularity needed by the stochastic approach, a straightforward solution method to obtain optimal pricing policies for real size systems is intractable.

Hence, to obtain pricing policies, we need to develop approximations. Chapters 3, 4 and 5 are devoted to three different solution methods, thus they can be read independently.

Chapter 3 deals with a scenario-based approach: a natural deterministic approximation. It amounts to optimizing a posteriori the system, considering that all trip requests (a scenario) are available at the beginning of the time horizon. Optimizing on a scenario provides heuristics and bounds for the stochastic problem. This approach raises a new constraint *the First Come First Served constrained (FCFS) flow*. We show that optimization problems based on FCFS flow are APX-hard. An approximation based on the MAX FLOW algorithm is investigated. MAX FLOW gives an upper bound on the scenario (offline) optimization and an approximation algorithm, with poor ratio, though useful to tackle the problem complexity.

Chapter 4 proposes an approximation algorithm to solve a simpler stochastic VSS pricing problem than the general one presented in Chapter 2. In order to provide exact formulas and analytical insights: transportation times are assumed to be null, stations have infinite capacities and the demand is Markovian stationary over time. We propose a heuristic based on computing a MAXIMUM CIRCULATION on the demand graph together with a convex integer program solved optimally by a greedy algorithm. For M stations and N vehicles, the performance ratio of this heuristic is proved to be exactly $N/(N + M - 1)$. Hence, whenever the number of vehicles is large compared to the number of stations, the performance of this approximation is very good. For instance, for systems with 15 vehicles per station on average ($N = 15M$), the performance guaranty is 14/16.

Chapter 5 is devoted to a fluid approximation of the stochastic process that can be seen as a plumber problem. The fluid model gives a static policy and an upper bound on the stochastic base model. This approximation has for advantage to consider time-dependent demands. The fluid heuristic policy will be shown to be the most efficient in practice.

The base stochastic model (of Chapter 2) is "hard" to evaluate exactly but easy to estimate through Monte Carlo simulations. Therefore, in Chapter 6, we propose a benchmark and a methodology to evaluate by simulation the pricing policies and bounds produced by our different solution methods. We consider for each policy its best fleet sizing that is computed by brute force. We show, under some assumptions, that pricing in vehicle sharing systems is a relevant optimization leverage.

Appendix A can be read totally independently from this thesis. Although the theoretical results presented were originally devoted to the study of the VSS stochastic discrete pricing problem, they have been generalized for a broader class of problems: continuous-time Markov Decision Processes (MDP) with large decomposable action spaces. This appendix intends to be pedagogic. A new quadratic program is proposed for general continuous-time MDP. A new linear programming formulation is presented for continuous-time MDP with decomposable action space. Finally, based on this linear program, we show that we are able to add constraints currently impossible to consider efficiently with state-of-the-art techniques.

(Introduction) En français

Contexte

Sur un échantillon de 22 villes aux états-unis, Shoup (2005) rapporte que les voitures à la recherche d'une place de parking contribuent pour 30% du trafic urbain. Par ailleurs, une voiture est utilisée en moyenne moins de 2 heures par jour ; elle occupe, le reste du temps, une place de parking. Pourrions nous avoir moins de véhicules tout en satisfaisant la même demande de transport ?

Récemment l'intérêt pour les systèmes de véhicules en libre service (VSS pour *Vehicle Sharing System* en anglais) a augmenté de manière significative. En effet, les politiques urbaines actuelles tendent à décourager les citoyens d'utiliser leurs véhicules personnels en limitant le nombre de places de parking, la taille des rues, etc. Les VSS semblent être un moyen prometteur pour réduire à la fois les embouteillages, le manque de places de parking, le bruit et la pollution (avec par exemple des vélos ou des véhicules électriques). Ils offrent une mobilité personnalisée en proposant à l'usager de ne payer que pour l'utilisation d'un véhicule, partageant ainsi son coût de possession.

Nous nous intéressons aux systèmes de véhicules en libre service en aller simple (*one-way VSS*) c'est à dire avec emprunt et restitution dans des lieux éventuellement différents. Associés aux réseaux de transport en commun classique, les *one-way VSS* permettent de résoudre le problème du dernier kilomètre (DeMaio, 2009). Cela n'est pas le cas lorsque l'on doit retourner son véhicule à la station où il a été emprunté (*round-trip VSS*).

Vélib' fut le premier VSS à grande échelle à proposer une location de vélos en aller simple. Il a été mis en place à Paris en 2007 et possède à présent plus de 1200 stations et 20 000 vélos, vendant environ 110 000 trajets par jour. Ce système a inspiré beaucoup d'autres villes à travers le monde. Aujourd'hui, plus de 300 villes possèdent des vélos en libre service tel Montréal, Pékin, Barcelone, Mexico, Tel Aviv (DeMaio, 2009).

Location en aller simple : un problème de gestion

La possibilité de louer un véhicule en aller simple (*one-way*) améliore la liberté de l'utilisateur mais au prix d'une plus grande complexité de gestion. Dans les systèmes en aller-retour (*round-trip*), lorsque l'on veut optimiser le rendement, le seul stock à considérer est le nombre de véhicules disponibles. Pour les *one-way VSS*, les véhicules ne sont plus la seule ressource clé : les stations de parking peuvent avoir un nombre limité de places, et les places libres deviennent alors également un mécanisme de contrôle important.

Depuis les premiers vélos en libre services, des problèmes de disponibilité de vélos et de place de parking sont apparus de manière récurrentes. Côme (2012), parmi d'autres, a utilisé des techniques de *data mining* pour analyser des données d'exploitation de VSS. Il a isolé des types d'usage pour comprendre la cause du déséquilibre de la répartition des vélos. Les raisons sont complexes mais nous pouvons cependant isoler deux phénomènes sous-jacents qu'il convient de maîtriser : "la gravitation" et "les marées". La gravitation implique que certaines stations sont chroniquement surchargées ou vides. Cela arrive par exemple pour les systèmes de vélos lorsque les utilisateurs rechignent à monter en haut d'une côte même s'ils se rendent dans cette zone (par exemple la butte Montmarte pour Vélib' à Paris). La marée est observable aussi bien pour les voitures que pour les vélos. Elle est due à de fortes demandes en direction et/ou en provenance des lieux de travail, commerciaux ou de loisir à des périodes précises de la journée ou de la semaine (métro-boulot-dodo ou ici VSS-boulot-dodo).

Pour améliorer l'efficacité des VSS, plusieurs perspectives ont été étudiées dans la littérature. Au niveau stratégique, certains auteurs se sont attaqués à déterminer la capacité et l'emplacement optimal des stations ; voir *e.g.* Shu *et al.* (2010), Lin and Yang (2011) et Kumar and Bierlaire (2012). Au niveau tactique, d'autres auteurs ont cherché à déterminer la taille optimale de la flotte de véhicules pour un ensemble de stations donné ; voir *e.g.* George and Xia (2011) et Fricker and Gast (2012). Au niveau opérationnel, pour atteindre une qualité de service raisonnable, dans la plupart des systèmes de vélos en libre service, afin de rééquilibrer les stations, des camions sont utilisés pour redistribuer les vélos ; voir *e.g.* Nair and Miller-Hooks (2011), Chemla *et al.* (2012), Contardo *et al.* (2012) et Raviv *et al.* (2013). L'objectif est de minimiser le nombre d'utilisateurs non servis : ceux désirant prendre un véhicule à une station vide ou en reposer un à une station pleine. Ce problème de redistribution revient à organiser des tournées de camions effectuant des prises et déposes de véhicules dans les stations.

Vers une autorégulation par incitation

Un nouveau type de VSS est apparu récemment : les voitures en libre services avec possibilité de location en aller simple tel Autolib' à Paris ou Car2go dans plus de 15 villes (Vancouver, San Diego, Amsterdam, Ulm...). À cause de la taille des voitures, l'optimisation opérationnelle avec des camions parait inapproprié. Une autre méthode d'optimisation du système doit être trouvé.

D'un point de vue expérimental, des politiques tarifaires heuristiques ont été étudiées par Chemla *et al.* (2013) et Pfrommer *et al.* (2013). Elles paraissent avoir une bonne performance dans leurs simulations. Cependant, elles ne fournissent aucun résultats analytique sur les potentiels gains d'une optimisation tarifaire. Fricker and Gast (2012) ont déterminé analytiquement la taille optimale de la flotte de véhicule dans des villes avec stations de capacités finies $\mathcal{K}$ et des demandes dites homogènes, c'est à dire que tous les trajets ont la même probabilité d'être effectués. Ils ont montré que, sans système de régulation, et même avec une taille de flotte optimale, chaque station de ces villes "parfaites" avait une probabilité de $\frac{2}{\mathcal{K}+1}$ d'être pleine ou vide. Fricker and Gast (2012) ont analysé une politique dynamique, "la puissance de deux choix" : lorsqu'un utilisateur arrive à une station, il donne au hasard deux stations possibles pour destination et le système le dirige systématiquement vers la moins chargée des deux. Pour leurs villes parfaites, ils ont montré que cette politique permet de réduire drastiquement la probabilité pour une station d'être pleine ou vide de $\frac{2}{\mathcal{K}+1}$ à $2^{-\frac{\mathcal{K}}{2}}$.

Dans cette thèse, nous étudions les leviers tarifaires d'optimisations pour obtenir une autorégulation du système. Grâce aux techniques de recherche opérationnelle, nous voulons estimer l'impact potentiel de l'utilisation de politiques tarifaires influençant les choix des utilisateurs afin d'obtenir des systèmes plus performant.

Résumé des contributions et présentation du manuscrit

L'objectif de cette thèse est d'étudier l'intérêt des politiques tarifaires pour optimiser les système de véhicules en libre service. La Figure 1 page 4 résume notre approche en donnant un aperçu des différents chapitres avec leurs dépendances. Nous détaillons maintenant rapidement les contributions de chaque chapitre.

Nous avons soulevé dans cette introduction une question informelle : est-il possible d'influencer les utilisateurs en jouant sur les prix des trajets afin d'améliorer significativement l'efficacité des *one-way VSS* ? Une idée intuitive de politique tarifaire est donnée dans la Figure 1 pour un système avec 3 stations et 8 véhicules. Cette politique est basée sur la disponibilité des *ressources*. Par exemple, la ressource véhicule est (très) disponible à la station "campus" et la ressource place de parking est également (très) disponible à la station "railway". Par conséquent, un trajet campus-railway est bon marché (\$). Cependant, le trajet inverse est cher (\$\$\$) car la ressource véhicule à la station railway et la ressource place de parking à la station campus sont rares.

Dans le Chapitre 1, nous donnons un aperçu général de l'optimisation dans les VSS. Nous présentons différentes facettes de la gestion des VSS et les leviers disponibles pour l'optimiser. Nous détaillons une terminologie pour l'optimisation tarifaire dans les VSS qui nous permet de présenter une revue de littérature classifiée.

Dans le Chapitre 2, nous proposons un modèle stochastique formel qui nous permet de définir le problème stochastique d'optimisation tarifaire des VSS. Ce problème est notre référence. Sa résolution est le "Graal" que nous poursuivons tout au long de cette thèse. Ce chapitre peut être lu indépendamment, bien que le Chapitre 1 permette de mettre en perspective les hypothèses et simplifications retenues. Même si ce modèle stochastique est simple, à cause de la granularité nécessaire à l'approche stochastique, une résolution directe pour obtenir des politiques tarifaires optimales pour des systèmes de tailles réelles est *intractable*.

Pour obtenir des politiques tarifaires, nous devons par conséquent passer par des approximations. Le but est de proposer des politiques heuristiques et des bornes afin d'identifier de potentiels gains d'optimisation. Les Chapitres 3, 4 et 5 sont dédiés à trois différentes méthodes de résolution, ils peuvent donc être lu séparément.

Lorsque l'on considère un problème stochastique, il est naturel de s'intéresser à ses variantes déterministes. Dans le Chapitre 3 la première approche que nous étudions est dite par scénario : toutes les demandes de trajets sont connues au début de l'horizon et le but est de trouver la politique statique qui maximise le nombre de trajets vendus pour ce scénario. L'optimisation d'un scénario procure une politique heuristique et une borne pour le problème stochastique. Cette approche soulève une nouvelle contrainte : *le flot premier arrivé premier servi* (flot FCFS). Nous montrons que les problèmes d'optimisation basés sur les flots FCFS sont APX-difficiles. Une approximation basée sur l'algorithme Flot Max est étudiée. Flot Max donne une borne supérieure sur l'optimisation d'un scénario et un algorithme d'approximation de performance théorique faible mais utile pour cerner la complexité du problème.

Dans le Chapitre 4, la deuxième approche traitée est la résolution avec garantie de performance d'un problème stochastique simplifié : une demande stationnaire, des stations de capacités infinies et des temps de transport nuls. Nous proposons une heuristique basée sur le calcul d'une Circulation Maximum sur le graphe des demandes couplé à un programme entier convexe résolu optimalement par un algorithme glouton. Pour M stations et N véhicules, le ratio de performance de cette heuristique est conjecturé être exactement $N/(N+M-1)$ et prouvé être au moins $(N-M)/(N+M)$. Par conséquent, lorsque le nombre de véhicules est grand devant le nombre de stations, la performance de cette approximation est très bonne.

Le Chapitre 5 présente l'étude d'une approximation fluide (déterministe) du processus markovien que l'on peut voir comme un problème de plomberie. Le modèle fluide produit une politique statique et une borne supérieure sur le modèle stochastique de base. Cette approximation a pour avantage de considérer des demandes dépendantes du temps. La politique heuristique fluide sera montrée par la suite comme la plus efficace en pratique.

Le modèle de base (du Chapitre 2) est “dur” à évaluer de manière exacte mais facile à estimer avec une simulation de Monte Carlo. C’est pourquoi, dans le Chapitre 6, nous proposons un benchmark et une méthodologie pour évaluer par simulation les politiques tarifaires et bornes supérieures produites par nos différentes méthodes de résolution. Nous considérons pour chaque politique, sa taille de flotte optimale par *brute force*. Nous montrons, sous certaines hypothèses, qu’une politique tarifaire adaptée est un levier d’optimisation pertinent pour les systèmes de véhicules en libre service.

L’Annexe A peut être lu indépendamment de cette thèse. Même si les résultats théoriques présentés étaient originalement dédiés à l’optimisation tarifaire d’un VSS par un modèle stochastique, ils ont été généralisé à une classe de problème plus large : les processus de décision Markovien à temps continu (CTMDP pour *continuous-time Markov Decision Process*) possédant un grand espace d’action décomposable. Cette annexe se prétend pédagogique. Nous proposons une formulation en programme quadratique pour les CTMDP. Une nouvelle formulation en programme linéaire est présentée pour les CTMDP possédant un espace d’action décomposable. Finalement, en se basant sur cette formulation linéaire, nous montrons que nous pouvons ajouter des contraintes jusqu’à présent impossible à considérer de manière efficace avec les méthodes actuelles.

Chapter 1

Vehicle sharing systems management

Science never solves a problem
without creating ten more.

George Bernard Shaw
(1856–1950)

Chapter abstract

The chapter gives a general overview of VSS management. We discuss different business models: decisions regarding the design choices and the offered services might involve a complex management of the system. The specificity of implementing a short term one-way VSS is presented. Strategical, tactical or operational optimization is often necessary to obtain descent performances. We situate the different decision levels and exhibit the link between each of them. It is the first brick to understand VSS management and where this thesis stands. We formally define a pricing framework for VSS studies. Thanks to this framework we are able to classify current literature results and exhibit where our contributions stand.

Keywords: One-way Vehicle Sharing Systems; Protocol; Strategical, tactical and operational optimization; Pricing policies classification; Literature review.

Résumé du chapitre

Ce chapitre procure un aperçu général sur la gestion des systèmes de véhicules en libre service. Nous discutons des différents business modèles : les décisions concernant la conception et les services offerts peuvent entrainer une gestion plus ou moins complexe du système. Nous expliquons les spécificités relatives aux systèmes avec possibilité de location en aller simple. Pour obtenir des systèmes performants, des optimisations stratégiques, tactiques et opérationnelles sont souvent nécessaires. Nous situons ces différents niveaux de décisions avec leurs inter connexions. C'est la première brique pour comprendre le contexte de cette thèse. Nous présentons un cadre formel pour l'optimisation des politiques tarifaires. Grâce à celui-ci une revue de littérature classifiée est présentée, permettant de situer nos contributions.

Mots clés : Véhicules en libre service ; Location en aller simple ; Protocole ; Optimisation stratégique, tactique et opérationnelle ; Classification des politiques tarifaires ; Revue de littérature.

Contents

1.1 Introduction

This thesis investigates pricing policies for optimizing Vehicle Sharing Systems (VSS). It is not clear whether pricing techniques can increase the system performances. To answer this question, we need to exhibit and quantify the specific interests of pricing in VSS. We have to understand what issues are present in VSS, and to put in perspective what pricing can achieve in comparison to other optimization leverages.

This chapter intends to give a general overview of VSS management. We discuss business model decisions regarding the offered services, the design choices, as well as strategical, tactical and operational optimizations. The aim is to situate the different decision levels and to exhibit the link between each of them. It is the first step to understand VSS management and where this thesis stands.

In Section 1.2, we present the specificity of implementing a short term one-way VSS. We give intuitions about some business model choice consequences such as the type of engine used for motorized vehicles (electric or gasoline). We discuss the different services to offer, and the way the user interacts with the system to access them, that we call *protocol*.

In Section 1.3, we review the different optimization leverages. We distinguish strategical, tactical and operational levels and give a brief literature review for each of them. We discuss the optimization criteria.

In Section 1.4, we formally define a pricing framework for VSS studies. Thanks to this framework we are able to classify current literature results and exhibit where our contributions stand.

1.2 VSS design, offered services & consequences

In this section we give a broad overview of VSS. We raise many questions that are presented in a descriptive manner. The aim is to exhibit the complexity of VSS management, to give insights to system's operators (decision makers) on the consequences of strategical, tactical, or operational decisions.

To illustrate this section, we consider two operators as examples for one-way car sharing systems: Autolib', that was inaugurated in 2011 in Paris, France, and Car2go, that was first pioneered in 2009 in Ulm, Germany, and that now operates over 7,300 vehicles, in 19 cities worldwide: Vancouver, San Diego, London, Vienna,... For bike sharing system examples, as of May 2011 there were about 375 around the world, we only refer to two of them: the first large scale one, Vélib', implemented in Paris in 2008, and the current largest one Hangzhou Public Bicycle in Hangzhou, China.

1.2.1 Design of one-way VSS

From round-trip to one-way rental The first car rental company, initially named "Rent-a-Car" and now known as Hertz, appeared in 1918 in Chicago, USA, with twelve Ford Model T cars. Car rental agencies were primarily dedicated to people who have a car that is temporarily out of reach or out of service. It is the case of travelers or owners awaiting the repair of their damaged vehicles. Another type of car rental users are those who like to change the type of their vehicle for a short period of time. In "classic" car rental, users pay by the day for a round-trip rental type, *i.e.* users have to return the vehicle at the same location they have taken it. Car rental is nowadays present everywhere around the globe.

Carsharing is a model of car rental where people regularly rent cars for short periods of time, often by the hour. This suits users that only need occasionally to use a vehicle. Carsharing operators can be commercial companies, coming from the classic car rental world such as Hertz on Demand by Hertz, or a dedicated firm such as Zipcar. These two companies are spread internationally, but there is also currently a trend for local cooperatives such as Cité lib in Grenoble, France. In carsharing systems, users have first to become members; then the reservation process is faster than classic car rentals. Cars can be rented by the hour but still have to be returned at the station where they were taken: a round-trip rental.

In this thesis we are interested in short-term one-way VSS in which vehicles can be taken and returned at different places paying by the minute. Associated with classic public transportation systems, short-term one-way VSS help to solve one of the most difficult public transit network design problem: the last kilometer issue (DeMaio, 2009). Round-trip VSS cannot address this important issue.

In bike sharing systems, to avoid competing with classic round-trip bike rentals that address another type of use, the rental price per hour increases rapidly with time. For instance in Vélib', the first half an hour is free, the second one costs 1 €, the third one 2 € and then it is 4 € for each half an hour.

Vehicle special features There are significant differences when dealing with bike or car sharing systems. Bikes are small and cheap: Bike Sharing Systems (BSS) are then easy to integrate in a city. It is possible to have a large fleet and numerous stations with large capacities. Vélib' Paris has 20 000 bikes among 1400 stations with 10 to 40 parking spots and Hangzhou Public Bicycle in China, the current biggest system, has 66 500 bikes among 2700 stations. However, with bikes, since users consider that it is "a citizen action" (green!) just to use the system, prices need to be low. Usually, it is free of charge for the first 30 minutes of use, and the annual subscription is in the order of a dozens Euro. The prices at stake are low and it seems hard to influence the demand with prices. To optimize the quality of services, trucks redistribution is commonly used to limit the number of empty/full stations. Bikes are small and trucks can carry from 20 to 50 bikes at the same time to rebalance the system.

Cars are bigger and more expensive; Car Sharing Systems (CSS) have usually less vehicles and stations with smaller capacities. For instance Autolib' has 1800 cars, 800 stations with capacity ranging from 1 to 6 parking spots. As CSS stations need more space than BSS ones, it can be an issue to install them in dense cities. Autolib' and Car2go have chosen small cars (respectively a Bluecar and a Smart) probably for this reason. Rental prices are higher in CSS since people are more willing to pay for renting a car than a bike. Pricing in this context should be a better leverage. An inconvenient of cars is that they need energy. An advantage is that they can be parked anywhere and can carry intelligent guiding systems (GPS).

Motorcycle sharing systems might be an interesting compromise between bikes and cars in term of price and size. However, motorcycles are more subject to accident which could be an issue!

Station/GPS-based systems There are three different system designs for VSS:

1. Station-based systems where vehicles are parked in specific stations with finite capacities: Users have to take and return their vehicle in these stations. This is the design of Autolib' and of usual BSS such as Vélib'.
2. GPS-based systems where vehicles can be parked in regular parking spots within a specific area: Users have to find them by GPS. This is the design of Car2go in London.
3. Mixed systems where there are dedicated parking spots across the city in addition to the possibility for users to park in regular ones. This is the design of Car2go in Vancouver.

One can notice that for modeling purpose, we can consider only the case of station-based systems. Indeed, a GPS-based system can be modeled as a station based one taking neighborhoods as stations with infinite capacities.

Energy management There are issues relative to the energy consumption of motorized vehicles. We distinguish two types of motor: gasoline engines that can be (almost) instantaneously recharged at several locations throughout the city, and electric engines that need a longer time and special conditions to be recharged at (currently few) specific spots.

First for gasoline CSS, there is usually a prepaid fuel card inside the car that users can use if they run short on gas. In Car2go the user is never required to fill up his vehicle. However, if the tank is less than 25% full before a user starts refueling, he receives 20 minutes of free drive time as a thank you. An advantage of gasoline CSS is that they do not need special recharging stations and can hence easily function in a GPS-based system.

Secondly, electric vehicles are more ecological, but imply a complex management. The recharge of batteries is subject to various constraints in order to optimize their performance and their life duration. Electric vehicles are probably simpler to manage in a station-based system. Though, one might think of a system in which batteries are charged in batch outside of the vehicles so that if a vehicle runs out of battery, the operator simply replaces it by a full one (Raviv, 2012).

In Car2go, the user has to return his vehicles at a recharging station (not all stations have this feature) when the battery load reaches a critical level. When a user ends his trip with less than 20% remaining in the battery, the car is automatically placed out of service until the fleet team is able to re-locate and charge the vehicle.

Dealing with electricity, the integration in the smart grid network might also be a subject of optimization. Pricing techniques can be applied to coordinate vehicle use and period of recharge to favor the use of electricity whenever it is cheaper.

1.2.2 Offered service – Rental protocol

When designing a one-way VSS, system operators need to decide what types of service to offer to the users: how the system can be used and what is the protocol for a user to access these services. In the following, we describe different possible services. They are not exclusive and they can be mixed to form a general offer.

Protocol definition A VSS user is interested in reaching a destination location d (GPS point), from an original location o (GPS point), during a specified time frame. For instance, for a morning commuter, o would be his home, d his work place, and his time frame would be constrained by leaving his bed not before 7am and by arriving at his desk not after 9am. To perform his journey, the user has to find an available vehicle close to his original location. He can question the system

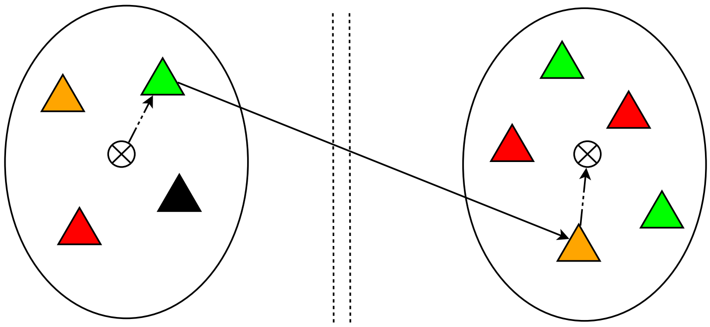


Figure 1.1: VSS protocol, from a GPS-GPS demand to taking a station-station trip.

either directly at a station or through a software (on his smart-phone for example). He has probably several admissible options to take his trip. These options respect his time frame, his time flexibility, but also his spatial and his price flexibilities. We call *protocol* the interaction, the communication process between the user and the system. If the protocol ends with a feasible solution, *i.e.* a spatio temporal trip admissible for the user and the system, the user takes this trip.

Figure 1.1 schemes an example of a station-based system in which a user wants to take a trip between two GPS locations (from left to right represented by the crosses). He has 4 admissible original stations and 5 admissible destination stations represented in the circles. He finally chooses a trip between 2 stations out of the 20 admissible trips. The way he reaches the original station (resp. destination location) from his original location (resp. destination station) can be hidden or not from the system's point of view. Hence, users might use different transportation means in coordination with the VSS system through a multi-modal trip planner. For instance, in Hangzhou Public Bicycle, the same card is used to take the bus and a bike.

Real-time rental To the best of our knowledge, in all BSS, users ask to take a vehicle to use right now. We call this protocol *real-time rental*. When a user takes a vehicle at a station, he can return it whenever and wherever he wants (under the condition of finding a free parking spot!). In station-based systems, due to the finite station capacities, it might be impossible for a user to return his vehicle at the desired station. It is a big issue for cars, first because costs are higher than for bikes and second because problems of traffic jams and blockings may occur. In Autolib', when a user is unable to find a free parking spot in a whole area, he can contact special agents to retrieve his vehicle. This management costs money and might be a bad experience for the user.

Parking spot reservation One of the solutions to avoid considering this "returning" problem is to introduce the possibility of reserving a parking spot at destination. For instance, Autolib' allows the user to reserve a parking spot at a station for 90 minutes. From the user's point of view, such reservation possibility is convenient. However, from the system's point of view, it might decrease the overall performance. Indeed, to ensure that a user will park his vehicle at a specified station, the system needs to lock a parking spot during the whole time of the trip. For high intensity rental systems, overbooking policies might significantly improve the number of trips. Nevertheless overbooking leads to a complex management of "collateral damages".

Trip reservation in advance Another feature that users could appreciate is the possibility to reserve a trip in advance. This possibility is offered in Autolib' and Car2go that allow to reserve a vehicle 30 minutes in advance. When receiving a future trip request from a station a to a station b, the system's unique way for being sure to serve it is: 1) to currently have a vehicle at station a and a free parking spot at station b, 2) to lock them until the reservation ends. Such reservation protocol (locking resources) may work when booking at most half an hour in advance or so. For earlier reservations, its efficiency might be poor because of its rigidity: a single reservation could lead to refuse a lot of trips.

Subscription Since some users are periodically taking the same trip, they might be interested to subscribe to a regular service. In this case, and maybe also for long term single trip requests, one could assume that users are ready to wait after expressing their request. During this period, the system could be able to consider several trip requests at the same time and to select which ones to serve in order to maximize a global interest.

Real-time hazards If a system's operator wants to offer reservations in advance or subscriptions, it is not reasonable to think that he will lock a vehicle and a parking spot for a couple of days until the trip happens. Moreover, if he coordinates several trip reservations to obtain a feasible (deterministic) solution, it could work in a theoretical world but may not meet the reality. Indeed, one cannot assume that all reservations will go through because of real-time hazards such as no shows, accidents, traffic jams... Hence, since problems of reservation feasibilities are inherent, allowing overbooking and considering real-time routing can be a solution to improve the system efficiency.

Overbooking When allowing reservations in advance, the system's operator needs to consider the probability of not going through a reservation of a vehicle or parking spot. It is also the case when playing with overbooking. In practice, special agreements with users, such as financial engagement or taxi use as substitution, should be considered.

Real-time routing When users have issues to find a free parking spot at destination, or if they have a car that is getting low in energy, the system operator might like to help them to route their vehicle in real-time in order to find a solution. Autolib' offers such service: when a user is unable to find a free parking spot in a whole area, he can contact special agents to retrieve his vehicle. Car2go maintains a "fleet team" that is dispatched to address issues such as low battery levels. Whenever any car falls below 20 percent charge, the fleet team is notified and a team member is dispatched to the car to bring it to a charging location.

Multi-modal routing VSS are part of the city public transportation system. Multi-modal trip planners, integrating VSS with other public transportation means, are interesting from the user's point of view. Regarding the VSS utilization maximization, such trip planner might also enable to dispatch the demand by offering alternative trips to users. Notice that for such trip planners, having systems offering trip reservations in advance could be interesting for users so they can really count on taking the VSS trip proposed.

Car pooling For CSS, cars' 5 or 7 seats might be seen as a resource to dispatch. To use efficiently the system, one should coordinate users taking the same trip, *i.e.* organizing car pooling. It would be a chance for users to diminish their individual financial cost as well as their ecological impact.

1.2.3 Understanding the demand

Data Mining Who is using VSS and what for? To answer this question, data mining analysis have been conduced in the literature. Many papers report the imbalances in the distribution of bikes on case studies; for instance Morency *et al.* (2011) on Montréal's BSS Bixi data and Vogel *et al.* (2011) on Vienna's BSS Citybike Wien data. Data Mining applied to operational data offers insights into typical usage patterns. It can be used to forecast demand with the aim of supporting and improving strategic and operational planning. Studies generally focus on station clustering analyses. Their goal is to find groups of stations with similar temporal usage profiles

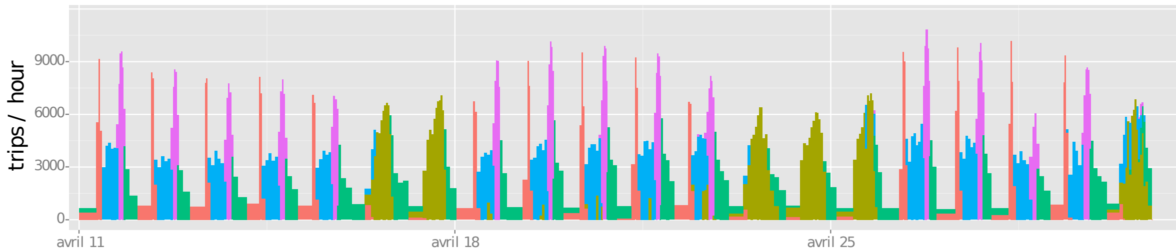


Figure 1.2: Spatio-temporal trip activity recognition on 3 weeks Vélib' data producing 5 clusters. Source Côme (2012).

(incoming and outgoing activity/hour). They usually report the same phenomenon: there are roughly two day patterns, a week day and a week-end day. For instance Côme (2012) partitions the stations in the following clusters: housing, employment, railway station, spare time, park and mixed usage. These clusters seem relevant: confronted to city economical and sociological indicators, it appears that without knowing the city, only looking at the exploitation data, Côme has been able to guess fairly well the housing/working areas, railway stations...

Côme (2012) develops also a spatio temporal trip activity recognition that he applies on Vélib'. Figure 1.2 shows such trip clustering on a 3 weeks horizon. Notice the similarity of the demand from one week to another and the similar pattern of week days and week-end days. Figure 1.3 represents the bike balance at Vélib' stations in the morning. Remark the separation into two types of station: with a clear positive or a clear negative balance. Vélib' stations have between 20 to 30 parking spots. Consequently, during these few hours, there is a flow voiding some stations and filling completely some others. This imbalance is the result of one of the spatio-temporal cluster identified by Côme, that he characterizes as a "house-work" demand. Together with the "evening opposite flow", the "work-home" cluster, we name this spatio-temporal phenomenon *tide*. Côme exhibits in total five clusters: house-work, lunch, work-house, evening and spare time.

Real demand estimation A decision maker may like to access the real demand. Transportation economists provide origin destination matrices from survey studies on mobility. However, these data reveal more macro than micro phenomenons, especially regarding new types of transportation systems. They do not give information directly usable to simulate a system. The only precise available data are the trips sold by the current system. The unserved demand is hidden, it is a problem of censored demand to build the real demand from the exploitation data. Anyway, even if rebuilt, the influence of new leverages such as pricing (demand elasticity) or

Figure 1.3: Spatial distribution of week day morning tide. Source Côme (2012).

reservation protocol are hard to estimate.

User behavior Another interesting information regards how users are making their decisions. What is the behavior of a user that cannot pick up or return a vehicle at a station? Is he waiting, going to the closest one or quitting? Is he interrogating, communicating, with a central system? What is the influence of prices on his decisions? To model a realistic VSS, we need to understand the links between price, spatial and temporal flexibilities. It is classic to consider an utility function linking the price to take a trip and an estimate cost of connections (*e.g.* by foot or public transportation) to reach the origin station and the destination location. Such function is an approximation, and its calibration can only be "heuristic". Moreover in practice, there are some threshold effects due to the competition of other transportation means such as taxi, subway, bus, bicycle...

1.3 VSS optimization overview

In this section, we review the different leverages of optimization and make a brief literature review for each of them. We categorize them by their level of decision. We start with strategical decisions with station location and size optimization. We continue with tactical decisions regarding the fleet sizing optimization. We end with the operational level of decision making. We talk about the vehicle balancing problem, the issues of reservations in advance and the use of pricing as incentive.

We finally discuss the importance of choosing a good criteria to optimize.

1.3.1 Strategical optimization: Station location & sizing

When implementing a VSS, one of the first questions is about the location and the capacity of its stations. This problem has been studied in the literature by several authors. Kumar and Bierlaire (2012) optimize the locations for a car sharing system in and around the city of Nice, France. The objective of their study is twofold: First, to analyze the performance of the car sharing service across all stations and estimate the key drivers of demand; Secondly, to use these drivers to identify future station locations, such that the overall system performance is maximized.

Other authors consider station locations joint with their optimal capacity. Shu *et al.* (2010) propose a stochastic network flow model to support these decisions. They use their model to design a bicycle sharing system in Singapore based on the demand forecast derived from current usage of the mass transit system. Lin and Yang (2011) consider a similar problem but formulate it as a deterministic mathematical model. Their model is aware of the bike path network and mode sharing with other means of public transportation.

Such theoretical studies have an interest to understand the VSS station location problem in general. However in practice, it appears that this problem might be more political than mathematical. Indeed, a station cannot be installed anywhere in a city and the implicit constraints necessary to understand these admissible locations are hard to formalize.

Ion *et al.* (2009) and Efthymiou *et al.* (2012) focus on people's perceptions (not the decision makers' one) about the appropriate location of mobility centers. Based on case studies, they propose an approach to model the preferences of the potential users. They intend to help the VSS managers, as well as the local authorities, to obtain an efficient management for existing systems and to study the opportunities of its extension.

1.3.2 Tactical optimization: Fleet sizing

At a tactical level, authors investigate the optimal number of vehicles (fleet size) given a set of stations. George and Xia (2011) study the fleet sizing problem with stationary demand and infinite station capacities. Fricker and Gast (2012), and later Fricker *et al.* (2012), consider the optimal sizing of a fleet in "toy" cities in which demand is homogeneous (*i.e.* stationary and identical for every trip), and where all stations have the same finite capacity $\mathcal{K}$. They show that even with an optimal fleet sizing in the most "perfect" city (that is homogeneous), without any operational system management, there is at least a probability of $\frac{2}{\mathcal{K}+1}$ for any station to be either empty or full.

Nair (2010) uses a network modeling framework to quantitatively facilitate design and operate VSS. At the strategic level, the problem of determining the optimal VSS configuration is studied.

In Vélib', the fleet size changes between summer and winter. According to the system's operator [1], winter and summer demands differ enough to the extent that changing the fleet size can increase the number of trips sold.

Notice that station sizing could also be seen as a tactical optimization in case of station with flexible capacity as in Bixi like systems.

1.3.3 Operational optimization: Vehicle balancing

At an operational level, in order to be able to meet the demand with a reasonable standard of quality, in most BSS, trucks are used to balance the bikes among the stations. The problem is to schedule truck routes to visit stations performing pickup and delivery. The objective is reset (rebalance) the system toward in its most efficient state, that is an input based on an ideal level of bikes filling for each station. In the literature, many papers deal with this problem. Raviv and Kolka (2013) study how to determine the best fill level of each station in a static repositioning setting. A static version of the BSS balancing problem is treated by Nair and Miller-Hooks (2011), Chemla *et al.* (2012) and Raviv *et al.* (2013). A dynamic version is tackled by Contardo *et al.* (2012) and Pfrommer *et al.* (2013). For a definition of the routing problems involved in the bike balancing problems, we refer to Chemla (2012).

1.3.4 Operational optimization: Reservation in advance

For real-time parking spot reservation (without overbooking), there are no issues regarding infeasibility. Kaspi *et al.* (2013) show that real-time reservation of parking spot at destination can improve the system performance under reasonable demand rates. For trip reservation in advance, real-time feasibility problems happen when a resource (vehicle or parking spot) is booked but is unavailable. Papier and Thonemann (2010) study a stochastic rental problem for a single station. They consider two classes of customer, one with reservation known in advance and one arriving stochastically. They show the dominance of a threshold policy.

1. JCDecaux has not communicated on the process of Vélib' fleet size optimization. It is probably based on empirical studies or on simulation.

For reservation in advance, one could assume that users are ready to wait after expressing their trip requests. During this period, the system is then able to consider several requests at the same time and to select which ones to serve in order to maximize a global interest. Putting apart real-time hazards, this problem becomes deterministic. For non flexible station to station requests, it is modeled and solved as a simple Max Flow in a time and space network. However, when considering GPS to GPS requests with time and space flexibilities, it can be reduced to a Max Flow With Alternative shown to be NP-hard in Section 3.6, page 78.

1.3.5 Operational optimization: Incentives/Pricing policies

Due to car sizes, operational balancing optimization through relocation with trucks seems inappropriate for car sharing systems. Moreover, for any type of VSS, balancing at night might be efficient but during the day time, using trucks downtown increases the city traffic to the extent that the benefits of this regulation system might be poor. Other ways of operational optimization have then to be found.

Without truck balancing, the only VSS regulation leverage is to act directly on the demand. One can assumes that it is possible to influence the demand with prices (incentives): basically higher the price is, lower the demand will be. The problem is then to decide which prices to set in order to optimize a global criteria. For instance, the demand can be considered continuous as schemed in Figure 1.4. The prices can range between giving money to users, to have the maximum demand Λ, and putting an infinite price, to "close" a trip. The space of feasible demand λ is then in $[0, \Lambda]$.

Pricing techniques seem easier to implement when reserving for a specified trip (vehicle and parking spot reservation); however with an appropriate communication system, it could also be used in other contexts. There are only few articles dealing with pricing in VSS; in Section 1.4 we propose a framework to classify and to situate them.

1.3.6 Optimization criteria

A complex decision Before optimizing anything, the criteria to consider have to be determined carefully. These criteria might depend on the referential. From the city's point of view, improving the mobility of the citizen is a consensual objective. But how to formulate mobility? An advantage of VSS is its ability to solve the last kilometer issue. Should we then prioritize the "forgotten" people of the current transportation systems? in the case of BSS, is it better to change the habit of a car commuter than to "convert a tramway user" for ecological reasons? When favoring

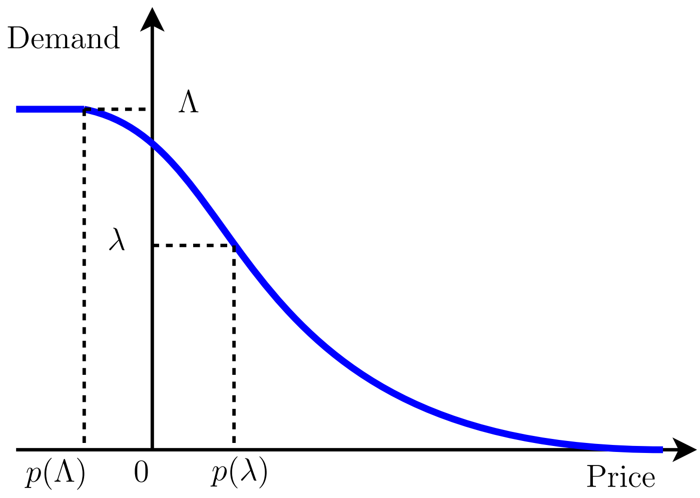


Figure 1.4: Continuous elastic demand $\lambda \in [0, \Lambda]$.

a user over another, some fairness considerations might be at stake. After all, taking an impartial criterion such as the number of trips sold or the vehicle utilization (both special cases of a more general revenue maximization criteria) might be the easiest ones to defend.

BSS case From BSS operators' point of view, the revenue generated by selling trips is usually low in comparison to what the system actually costs. According to Midgley (2011), capital costs can be up to \$4,500 per bicycle, and annual operational costs up to \$1,700 per bicycle. Indeed in most BSS business models, the first half an hour is free of charge and the annual subscription to use the system is in the order of a dozen Euro. Operators are earning the majority of their revenue by a third activity such as advertisement or public funding. For instance, the city of Paris has given to Vélib' operator JCDecaux all the public advertising panels (bus stop...) of the French capital in exchange for providing a BSS to the Parisian (CRC, 2012). In Autolib' car sharing system, rental prices are a little bit higher than BSS but don't cover the expenses either. In this case, as explained by Jacqué (2013), it is a way for Bolloré, the system's operator, to promote his electric battery installed in the shared vehicles (Bluecar).

Operators don't have a direct interest to optimize their system because it is unclear for them whether they will have a return on their investment. Therefore, cities use to contract them to ensure a minimum quality of services. In this case, the optimization's objective has a threshold form, for a given indicator. Even if there are no perfect indicators, for a matter of measurable simplicity, one might be tempted to choose bad ones! For instance, as stated in CRC (2012) [2], an indicator chosen for Vélib' is the average percentage of time a station is in a problematic state (empty or full). At first sight, this indicator seems reasonable, but in fact it doesn't lead to maximize the number of trips sold by the system (another indicator maybe more objective). For instance, when the city has a strong morning and evening tide from a Home area (H) to a Work area (W), the policy that maximizes the number of trips sold can lead to have all stations in H full and all station in W empty at night, and the opposite during the day. In this policy, the stations are characterized as being in a problematic state most of the day even though it is maximizing the system utilization.

Transit maximization One problem with pricing incentives is the hardness to make explicit the elasticity function (linking price and demand). It can be a complex function (not continuous, with thresholds...). Moreover, setting the proper prices to obtain a fixed (optimized) demand might require the skills of an economist and experimental studies. On the other hand, there exist some objectives such as maximizing the number of trips sold (transit) or the total travel time that do not need an explicit elasticity function. The only data necessary for such optimization is the space of the possible demand, for instance $\lambda \in [0, \Lambda]$ for continuous elastic demand. With such assumptions, prices become implicit and pricing policies can be seen as incentive policies or simply policies regulating demand.

For these reasons, most of the results in this thesis focus on the transit optimization. The first question one might raise is whether it is possible to improve on the number of trips sold by the *generous pricing policy* that is accepting the maximum potential demand (for every trip (a, b) the generous pricing policy sets the demand $\lambda_{a,b} = \Lambda_{a,b}$). Let us explain why this question is not trivial. For a given pricing/incentive policy λ, for each trip (a, b) we distinguish between the potential demand $\lambda_{a,b} \in [0, \Lambda_{a,b}]$ and the satisfied demand $y^{\lambda}_{a,b} \in [0, \lambda_{a,b}]$ (the average flow of users served). This difference is schemed in Figure 1.5. The question amounts to finding a pricing policy λ such that $\sum_{a,b} y^{\lambda}_{a,b} > \sum_{a,b} y^{\Lambda}_{a,b}$.

2. A report on Vélib' management ordered by the city of Paris.

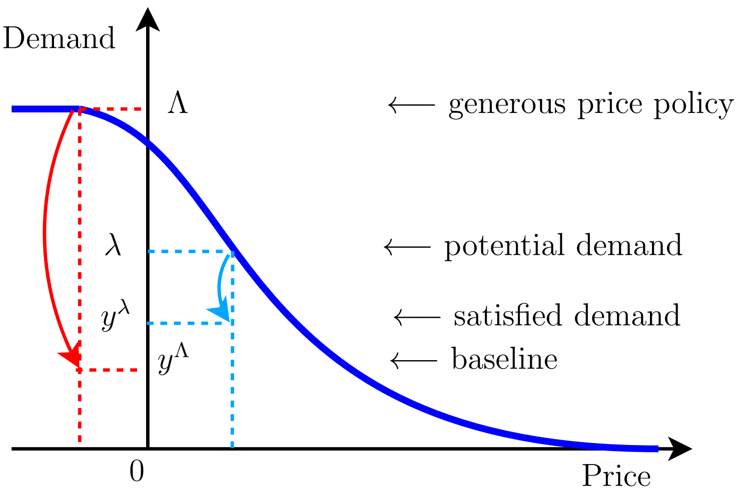


Figure 1.5: Can pricing improve on the transit of the generous policy?
$\Leftrightarrow$ $\exists$? pricing policy λ such that $\sum_{a,b} y^{\lambda}_{a,b} > \sum_{a,b} y^{\Lambda}_{a,b}$.

1.4 VSS pricing optimization

1.4.1 Revenue management in vehicle rental system

The origin of Revenue Management (RM) lies in the airline industry. It started in the 1970s and 1980s with the deregulation of the market in the United States. In the early 1990s RM techniques were then applied to improve the efficiency of round-trip Vehicle Rental Systems (VRS); see Carroll and Grimes (1995) and Geraghty and Johnson (1997). One-way rental is now offered in many VRS but usually remains much more expensive than round-trip rental. One-way VRS RM literature is recent. The closest paper we refer to is Haensela *et al.* (2011) who model a network of round trip car VRS with the possibility of transferring cars between rental sites for a fixed cost.

For trucks rental, companies such as Rentn'Drop in France or Budget Truck Rental in the United States are specialized in one-way rental, offering dynamic pricing. This problem is tackled by Guerriero *et al.* (2012) who consider the optimal management of a fleet of trucks rented by a logistic operator. The logistic operator has to decide whether to accept or reject a booking request and which type of truck should be used to address it.

Results for one-way VRS are not directly applicable to VSS, because they differ on several points: 1) Renting are by the day in VRS and by the minute in VSS with a change of scale in the temporal flexibility; 2) One-way rental is the core of VSS; for instance round trip rental represents only 5% of sold trips in Bixi (Morency *et al.*, 2011), whereas it is classically the opposite in car VRS; 3) There is usually no reservation in advance in VSS, it is a “first come first served rule”, whereas usually trips are planned several days in advance in VRS. At the best of our knowledge there are no results in the RM literature dedicated to pricing in VSS.

1.4.2 Pricing policies classification

We distinguish between different class of pricing policies. Classic pricing policies take into account only the rental length and traveled distance. But apart from adjusting the offer and demand to optimize an overall revenue, they give no operational leverage on the system. To drive the system towards its most profitable state, we need to influence the users by considering their diversities. With a parking spot reservation protocol, the system knows exactly which trip a user wants to take. We define two types of pricing policy, by station or by trip.

Definition 1 (Station Pricing). *A station pricing policy sets for each station a price to take a vehicle and a price to return it. More formally, the price $p_{a,b}(t_a, t_b)$ to take a trip from station a at time t_a to station b at time t_b is a function P of the classical price $c_{a,b}(t_a, t_b)$ to take such trip and the incentives $tak_a(t_a)$ to take a vehicle at station a at time t_a and $ret_b(t_b)$ to return it at station b at time t_b:*

$$p_{a,b}(t_a, t_b) = P(tak_a(t_a),\ ret_b(t_b),\ c_{a,b}(t_a, t_b)).$$

Definition 2 (Trip Pricing). *A trip pricing policy sets a price to take each trip. More formally, the price $p_{a,b}(t_a, t_b)$ to take a trip from station a at time t_a to station b at time t_b is a function P of the classical price $c_{a,b}(t_a, t_b)$ to take such trip and the incentive $trip_{a,b}(t_a, t_b)$ to take the trip (a, b) between t_a and t_b:*

$$p_{a,b}(t_a, t_b) = P(trip_{a,b}(t_a, t_b),\ c_{a,b}(t_a, t_b)).$$

Incentives can depend or not on the system's state, *i.e.* the vehicle distribution among the stations and the trip routes. It defines two types of policies: static and dynamic.

Definition 3 (Static Pricing). *A static pricing policy sets (in advance) a price to take every trip at any time independently of the system's state.*

Definition 4 (Dynamic Pricing). *A dynamic pricing policy can set a price to take a trip that depends on the current state of the system.*

Operators have to consider the KISS principle (Keep It Simple and Stupid) in order to have their optimized policies accepted by users. They might be interested in a simple class of policies, easier to conceptualize for them, for the user as well as for the optimizer. We define a simple class of dynamic policies in which prices depend only on the current state of the pick up and return stations, and a simple class of static policies in which prices do not change along the day.

Definition 5 (Locally Dynamic Pricing)**.** *A station state dependent pricing policy can set the price to take a trip from a station a to a station b in function of the current states of stations a and b (parking filling and number of vehicles in transit toward them).*

Definition 6 (Fully Static Pricing)**.** *A fully static pricing policy is a static policy that is constant over time, i.e. prices depend neither on the system's state nor on the time the trip is taken.*

When studying pricing policies, the demand is usually considered elastic in function of the proposed price. It means roughly that higher the price is lower the demand will be. In theory, with perfectly rational users we would have a bijective function linking prices and demands and therefore consider only continuous prices. However, because of psychological or marketing issues, we can be interested in studying discrete prices.

Definition 7 (Continuous Pricing)**.** *The price p to take each trip has to be selected in a range: $p \in [p_{\min},\ p_{\max}]$.*

Definition 8 (Discrete Pricing)**.** *The price p to take each trip has to be selected in discrete values: $p \in \{p_1,\ \ldots,\ p_k\}$.*

With such formal definitions, we are now enable to categorize the studies done in the literature.

1.4.3 Classified literature review

Only few studies have been conduced in the literature to study the influence of pricing in VSS. Papanikolaou (2011) proposes ideas for locally dynamic station pricing based on the system dynamics framework. His model can be used as an educational tool for studying the behavior of VSS and exploring the impact of pricing policies on transit optimization. Pricing concepts are developed but no experimental results are provided.

Chemla *et al.* (2013) implement a simple locally dynamic continuous station pricing heuristic based on Monge's transportation problem. Prices are updated regularly and aim at deterring users from parking at stations that are already nearly full. Users have incentives to park at other stations that have a greater number of available parking places. Taking a sample of cities generated randomly, they show that adding a pricing regulation improves significantly the level of service.

Fricker and Gast (2012) study the impact of a simple dynamic policy in homogeneous cities. It is based on the power of two choices paradigm: the user indicates randomly two destination stations and he is routed to the least loaded one. They show that it improves dramatically the situation: with an optimal sizing the probability for a station to be empty or full decreases from $\frac{2}{\mathcal{K}+1}$ to $2^{-\mathcal{K}}$ with $\mathcal{K}$ the uniform station capacity. However, their results concern perfectly balanced cities (homogeneous) and studies on more realistic cites are not investigated yet.

Pfrommer *et al.* (2013) consider a combination of intelligent repositioning decisions and dynamic pricing. Based on model predictive control principles, they develop a locally dynamic discrete station pricing with 10 different prices for the destination station (no incentive for the pick up one). They want to encourage customers to change their destination station in exchange of a payment to improve the overall service level.

1.4.4 Pricing in practice

Vélib' has developed a fully static station pricing. Some "stations +" are characterized by their high elevation such as Montmartre (hill). To reduce the gravitation phenomenon, when a user returns a bike at a "station +", he receives in exchange 15 minutes of free ride that he can accumulate to execute a longer journey without paying (recall that the first half an hour is free).

Figure 1.1, page 19, gives an example of what could produce a discrete pricing per station. A color code is used to represent the different discrete prices to take and to return a vehicle. The green, the yellow and the red triangles represent cheap, normal and expensive stations respectively. The black triangles represent unavailable stations for taking on the left part, or returning on the right part. In this example, the user has finally chosen to pay for a "green taking" and a "yellow returning".

Figure 1.6 shows a graphical user interface of a discrete pricing per station for GPS based system (in this case it could have been named a discrete pricing per location). The colors on the area represent different price levels.

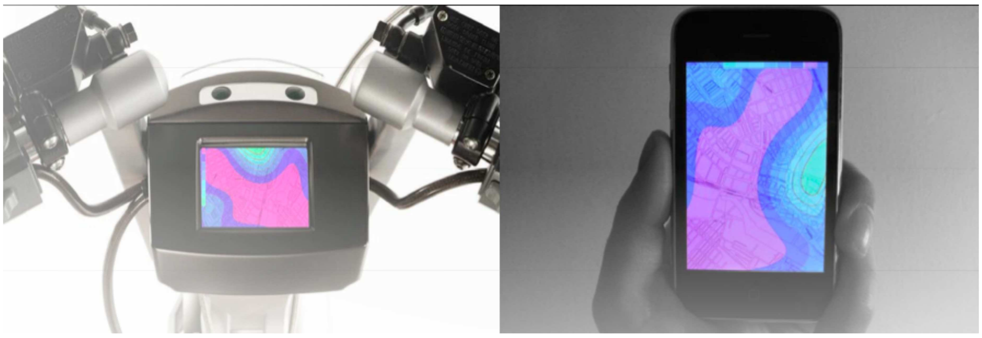


Figure 1.6: Price heated map. Source MIT Media Lab (Papanikolaou, 2011).

1.4.5 Digressions on pricing psychological impact

For station pricing policies, at least two types of price function exist to compute the incentives: (1) *additive incentive* adding a fixed fee to the classical renting price and (2) *multiplicative incentive* multiplying the classical renting price by an incentive factor.

We conjecture that considering only additive incentives will impact mainly small distance trips. For instance, if a trip from a to b has a negative incentive of +5€ to take a vehicle in station a, and no incentive (+0€) to return it in station b, this overall incentive of +5€ can be considered as substantial for a 15€ trip but barely unnoticeable for a 70€ trip.

On the contrary, considering only multiplicative incentives will impact mainly the long distance trips. For instance, for a trip from a to b which has a negative incentive of +10% to take the vehicle in station a, and no incentive (+0%) to return it in station b: this overall incentive of +10% has probably almost no effect on a 15€ trip (1.5€) whereas on a 70€ trip, this 7€ difference might represent more for the potential user.

Therefore, to tackle both short and long trips, the best pricing would be a combination of multiplicative (α) and additive (β) incentives. Hence, a reasonable set of discrete station prices to take/return a vehicle could be the following couples (α, β): (+0€,+0%), (+2€,+3%) and (+5€ ,+5%).

Finally, a user going from a station a, with incentive $(\alpha^a_{tak}, \beta^a_{tak})$ to take a vehicle, to station b, with incentive $(\alpha^b_{ret}, \beta^b_{ret})$ to return a vehicle and for a renting time T costing $c(T)$, will have to pay:

$$p(a, b, T) = \alpha^a_{tak} + \alpha^b_{ret} + (\beta^a_{tak} + \beta^b_{ret}) \times c(T).$$

In the end, these digressions are related to the economical and psychology field. In the mathematical models developed in this study, we assume that users are perfectly rational and do not consider explicitly the economical incentives.

1.5 Our contribution: pricing studies

In this thesis we focus on VSS pricing to optimize the number of trips sold. To the best of our knowledge, only heuristics have been studied in the literature for this problem. There are neither mathematical models nor solution methods to estimate the potential optimization gap of pricing. This is the scope of this thesis.

- In Chapter 2 we give a formal definition of VSS stochastic pricing problems. We propose a Markov decision process to model the dynamic discrete trip pricing. This exact model is intractable for real-world instance optimization though useful theoretically.
- In Chapter 3 we study a deterministic pricing problem to tackle static trip and station pricing. It is an offline optimization that provides an upper bound on a stochastic realization.
- Chapter 4 is devoted to the study of a simplification of the VSS stochastic problem. An approximation algorithm and an upper bound is given for the dynamic continuous trip pricing.
- Chapter 5 tackles a fluid approximation of the broader stochastic pricing problem studied. We provide a static continuous pricing heuristic policy and a conjectured upper bound on dynamic pricing optimization.
- In Chapter 6 we propose methodology to compare by simulation the different leverages of optimization. We propose a benchmark and exhibit the interest of pricing.

Chapter 2

A VSS stochastic pricing problem

Everything should be made as simple as possible, but not simpler.

Albert Einstein (1896–1955)

Chapter abstract

This chapter presents the VSS stochastic pricing problem. This problem is our reference, the "Holy Grail" that we try to solve all along this thesis. It focuses on a simple real-time station-to-station reservation protocol. For a given pricing policy, the VSS dynamic is modeled as a closed queuing network: the VSS stochastic evaluation model. It allows to give a formal definition of the VSS stochastic pricing problem. An exact measure of the VSS stochastic evaluation model is intractable for real size systems. Hence, solving in general the VSS stochastic pricing problem appears hard. We discuss notions of complexity in this stochastic framework.

Keywords: Modeling; Stochastic process; Continuous-time Markov Chain; Markov Decision Process; VSS Stochastic Pricing Problem; VSS Stochastic Evaluation Model; Optimal policies characterization.

Résumé du chapitre

Ce chapitre présente le problème stochastique de tarification dans les systèmes de véhicules en libre service. Ce problème est notre référence,

sa résolution est le "Graal" que nous poursuivons tout au long de cette thèse. Il considère un protocole simple, des requêtes en temps réel entre deux stations avec réservation d'une place à la station de destination. Pour une politique tarifaire donnée, la dynamique de ce système peut être modélisé par un réseau de files d'attente fermé : le modèle stochastique d'évaluation. Cela nous permet de donner une définition formelle du problème stochastique de tarification. Une mesure exacte du modèle stochastique d'évaluation est intractable pour des systèmes de taille réelle. Résoudre ce problème de manière général parait dure. Nous discutons de notions de complexité dans cet environnement stochastique.
Mots clés : Modélisation ; Processus stochastique ; Chaine de Markov à temps continu ; Processus de décision Markovien ; Modèle stochastique d'évaluation ; Problème stochastique de tarification ; Caractérisation de politiques optimales.

Contents

2.1 I model – You model (– God models) – Math model

This research intends to answer an informal question: is pricing a relevant leverage in one-way VSS management? As Mathematicians, we can only answer to questions that are formalized. Therefore, to use mathematics in real-world problems we have to go through a modeling phase. A model is by definition imperfect [1];

1. A scientific model is an approximation of a real system that omits all but the most essential variables. According to http://www.thefreedictionary.com, a model is a small object, usually built

the relevant question is whether a model catches interesting features, so that solving it gives useful knowledge on the original (real world) problem?

Because models only catch part of the problem, solutions have to be understandable in order to derive *practical theories*, *i.e.* reliable for a decision maker in the real-world context. A classic rule of thumb, based on Ockham's razor [2], is to isolate subproblems and to keep the model as simple as possible. Such decomposition is based on the belief (*faith*?) that one can find, in every complex process, a structure which can be applied generally. In other words, solutions with such structure [3], if not dominant, will have a good performance in a broader class of problems.

For all these (good) reasons, we propose in this chapter a stochastic optimization model, simple on purpose, that we hope will help to derive practical theories about pricing in real world VSS.

A stochastic problem VSS dynamic is random in nature. When dealing with human behavior, the variabilities of user arrivals and transportation times are high. Deterministic models are unlikely to include these uncertainties. In this context, stochastic optimization seems the most relevant approach to cope with randomness.

Stochastic models are used in several fields of research such as traffic flow, game theory, queueing networks, reliabilities, epidemic spreading or finance. There is a wide range of books dealing with stochastic processes. If the reader is not familiar with this area, we refer to Wang (2001). She discusses how to learn stochastic processes and she gives an overview of the literature with its most common mathematical techniques.

2.2 A VSS Stochastic Model

2.2.1 A real-time station-to-station reservation protocol

In a real-life context, a user wants to use a vehicle to take a trip between an original (GPS) location a, and a final one b, during a specified time frame. In a station based VSS, he tries to find the closest station to location a with a vehicle available and the closest station to location b with a free parking spot to return

to scale, that represents in detail another, often larger object. It is a schematic description of a system, theory, or phenomenon that accounts for its known or inferred properties and may be used for further study of its characteristics.

2. Ockham's razor is a principle of parsimony, economy, or succinctness used in logic and problem-solving. It states that among competing hypotheses, the hypothesis with the fewest assumptions should be selected.

3. With possibly minor changes.

it. All along this process, user's decisions rely on several correlated inputs such as: trip total price, walking distance, public transportation competition, time frame... A time elastic GPS to GPS stochastic demand, linked to a user's behavior decision protocol choosing its origin/destination stations and time frame seems closer to reality but introduces complexity (the use of utility function for instance).

In this study, we consider a simple *real-time station-to-station reservation protocol* as defined in Figure 2.1. When a user takes a vehicle he commits to return it at a specified time to a destination station. In counterpart, the system reserves him a parking spot to ensure the feasibility of his trip. This protocol simplifies the system specification; for instance it does not need to define the user's behavior when trying to return a vehicle at a full station.

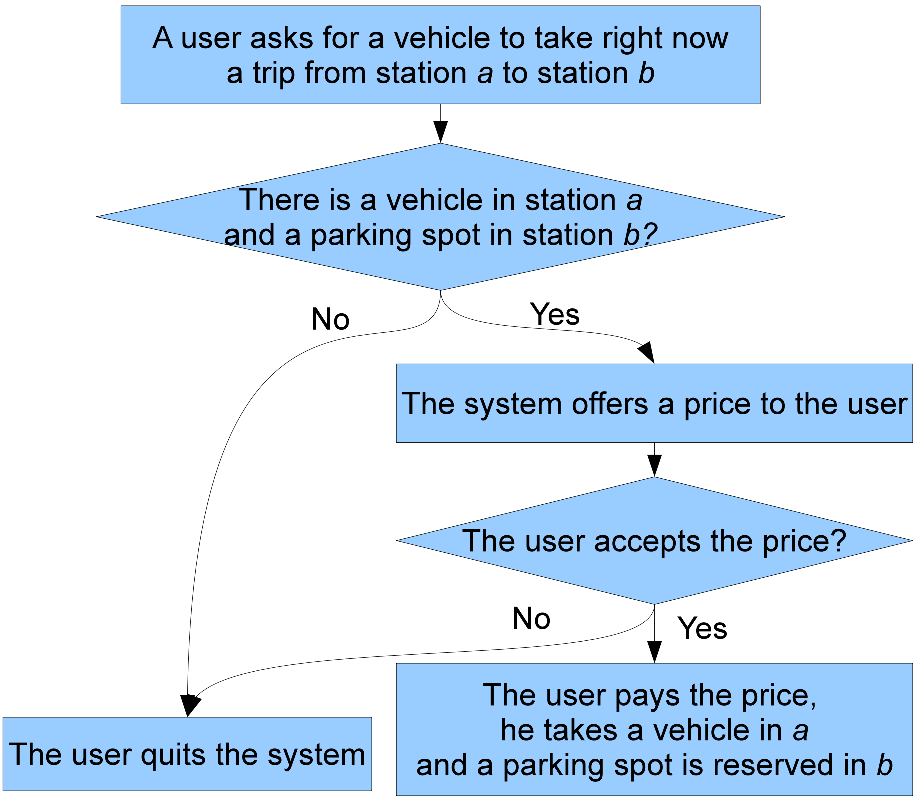


Figure 2.1: The real-time station-to-station reservation protocol.

2.2.2 An implicit pricing

We now recall definitions and assumptions on the prices discussed Chapter 1.

Concept of maximum potential demand We assume that for each trip (a, b) and independently of the other trips, there is a pool of potential users that may try to take trip (a, b) in the time horizon of the model.

Pricing policies and incentives We assume that there exist leverages (incentives) able to decrease the maximum demand (separately for each trip). A classic incentive is the price to take a trip; the demand is then a function of the price: basically, the higher the price, the lower the demand.

A pricing/incentive policy is *static* if the price to take each trip is independent of the state of the system. A policy is *dynamic* otherwise. Prices can be either *discrete*, implying a discrete set of possible demand, *i.e.* selected in a set of values, or *continuous*, *i.e.* chosen in a range.

Continuous elastic demand We assume the following hypothesis for continuous pricing optimization: Let $\Lambda^t_{a,b}$ be the maximum demand of users who want to take a trip at time step t between stations a and b. There exists a price $p(\lambda^t_{a,b})$ to obtain any demand $\lambda^t_{a,b} \in [0, \Lambda^t_{a,b}]$. A price function is schemed Figure 2.2. Notice that, in this example, the maximum demand Λ is obtained with a minimum price $p(\Lambda)$ that is negative. Indeed it is conceivable that the system chooses to pay users to take certain trips (instead of paying trucks).

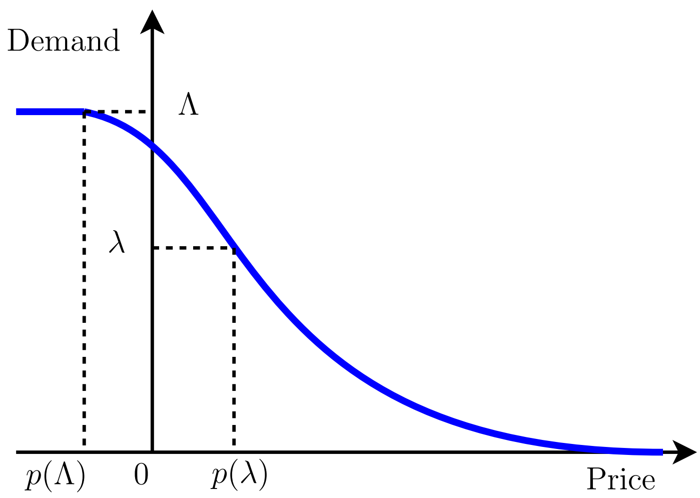


Figure 2.2: Continuous elastic demand $\lambda \in [0, \Lambda]$.

Implicit pricing One problem with pricing incentives is the hardness to make explicit the elasticity function (linking price and demand). It can be a complex function (not continuous, with thresholds...). Moreover, setting the proper prices to obtain a fixed (optimized) demand might require the skills of an economist and experimental studies. On the other hand, there exist some objectives such as maximizing the number of trips sold (transit) or the total travel time that do not need an explicit elasticity function. The only data necessary for such optimization is the space of the possible demand, for instance $\lambda \in [0, \Lambda]$ for continuous elastic demand. With such assumptions, prices become implicit and pricing policies can be seen as incentive policies or simply policies regulating demand.

For these reasons, most of the results in this thesis focus on the transit optimization. The first question one might raise is whether it is possible to improve on the number of trips sold by the *generous pricing policy* that is accepting the maximum potential demand (for every trip (a, b) the generous pricing policy sets the demand $\lambda_{a,b} = \Lambda_{a,b}$). Let us explain why this question is not trivial. For a given pricing/incentive policy λ, for each trip (a, b) we distinguish between the potential demand $\lambda_{a,b} \in [0, \Lambda_{a,b}]$ and the satisfied demand $y^{\lambda}_{a,b} \in [0, \lambda_{a,b}]$ (the average flow of users served). This difference is schemed in Figure 2.3. The question amounts to finding a pricing policy λ such that $\sum_{a,b} y^{\lambda}_{a,b} > \sum_{a,b} y^{\Lambda}_{a,b}$.

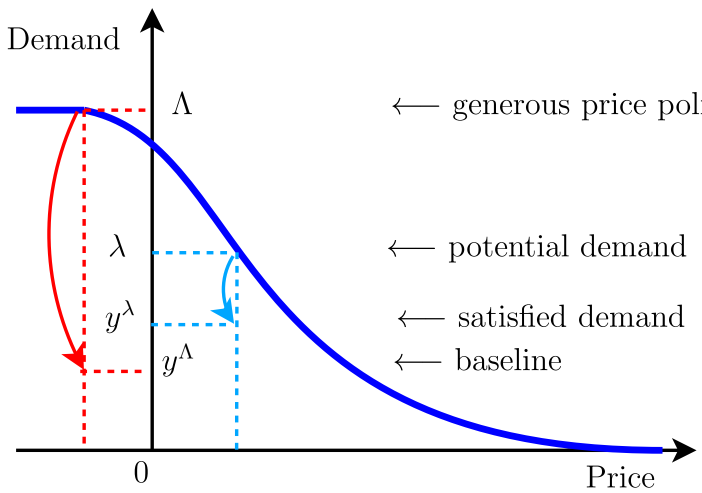


Figure 2.3: Can pricing improve on the transit of the generous policy?
$\Leftrightarrow$ $\exists$? pricing policy λ such that $\sum_{a,b} y^{\lambda}_{a,b} > \sum_{a,b} y^{\Lambda}_{a,b}$.

Finally, one interest of having an explicit formulation of the demand elasticity function is to maximize the revenue of the system. However, solving the revenue induces non-linearities in the optimization model. While the transit maximization (or any other linear function in y^{λ}, such as the maximization of "the total travel time" or "the total gain of travel time by using the system"...) leads to linear optimization models. Avoiding non-linearities (computational complexity) is therefore another reason to focus on transit maximization.

2.2.3 The VSS stochastic evaluation model

Continuous-time Markov chain evaluation framework We model the VSS dynamic by a stochastic process: the *VSS stochastic evaluation model*. This model does not consider any decision, it only measures/evaluate VSS performances for a given policy (demand vector). We use this evaluation model to compare the performance of policies using leverages such as the prices (regulating demands), the fleet sizing, the station capacities... We now define formally the VSS stochastic evaluation model under the real-time station-to-station reservation protocol (defined in Figure 2.1). We assume that all durations follow exponential distributions, therefore VSS dynamic becomes Markovian and a policy λ can be modeled as a continuous-time Markov chain.

VSS Stochastic (Markovian) Evaluation Model

- **Input:**
  - A number N of vehicles;
  - A set $\mathcal{M}$ of stations with capacities $\mathcal{K}_a,\ a \in \mathcal{M}$;
  - A set $\mathcal{T}$ of time steps with mean duration $\tau^t,\ t \in \mathcal{T}$, the horizon is periodic with mean total duration $T = \sum_{t \in \mathcal{T}} \tau^t$;
  - The mean of the transportation times duration $1/\mu^t_{a,b}$ for every trip $(a,b) \in \mathcal{D} = \mathcal{M} \times \mathcal{M}$ at every time step $t \in \mathcal{T}$;
    - A set $\mathcal{S}$ of states:

$$\mathcal{S} = \left\{ \Big(n_a \in \mathbb{N} :\ a \in \mathcal{M},\ n_{a,b} \in \mathbb{N} :\ (a,b) \in \mathcal{D},\ t \in \mathcal{T}\Big) \right. \\ \left. /\ \sum_{i \in \mathcal{M} \cup \mathcal{D}} n_i = N \ \&\ n_a + \sum_{b \in \mathcal{M}} n_{b,a} \leq \mathcal{K}_a,\ \forall a \in \mathcal{M},\ \forall t \in \mathcal{T} \right\};$$

    - A state $s = (n_a :\ a \in \mathcal{M},\ n_{a,b} :\ (a,b) \in \mathcal{D},\ t \in \mathcal{T})$ represents the vehicle distribution in the city space (in station or in transit) at a given time: At time step t, n_a is the number of vehicles in station $a \in \mathcal{M}$ and $n_{a,b}$ is the number of vehicles in transit between stations a and b serving a trip demand $(a,b) \in \mathcal{D}$.
    - The arrival of a vehicle at station b from station a is represented by a transition rate $n_{a,b}\mu^t_{a,b}$ between states $(\dots,\ n_b, \dots,\ n_{a,b} \dots,\ t)$ and states $(\dots,\ n_b + 1, \dots,\ n_{a,b} - 1, \dots,\ t)$ with $n_{a,b} \geq 1$;
    - The changing between two piecewise constant demand time steps is represented by a transition rate $1/\tau^t$ between states $(\dots,\ t)$ and states $(\dots,\ t + 1 \bmod |\mathcal{T}|)$.
  - A policy λ:
    - $\lambda^s_{a,b}$ is the arrival rate of users to take trip $(a,b) \in \mathcal{D}$ between states $s = (\dots,\ n_a, \dots,\ n_{a,b} \dots,\ t)$ and states $(\dots,\ n_a - 1, \dots,\ n_{a,b} + 1, \dots,\ t)$ with $n_a > 0$ and $n_b + \sum_{c \in \mathcal{M}} n_{c,b} < \mathcal{K}_b$;
    - The continuous-time Markov chain defined by states $\mathcal{S}$ and transition rates $\lambda,\ \mu$ and τ^{-1} is supposed to be strongly connected.
- **Output:** Indicators on the steady state behavior of the continuous-time Markov chain defined by states $\mathcal{S}$ and transition rates $\lambda,\ \mu$ and τ^{-1} such as:
  - The expected number of trips sold;
  - The expected vehicle utilization.

Notice that to measure the expected revenue, the price to take each trip would need to be specified in the input (a function $\lambda^t_{a,b}(s) \mapsto p^t_{a,b}(s)$).

The number of states of the continuous-time Markov chain is exponential in the number of vehicles and stations. For instance, for one time step, without transportation time and with infinite station capacities there are $\binom{N+|\mathcal{M}|-1}{N}$ states (Proposition 1). It means that for a system with $N = 150$ vehicles and $|\mathcal{M}| = 50$ stations, there are already about 10^{47} states!

Proposition 1. *The number of state of the Markov chain for N vehicles and M stations with infinite station capacities and null transportation time is equal to* $\binom{N+M-1}{N}$.

Proof. The states of the Markov chain for N vehicles and M stations are in one to one mapping with non decreasing functions from $\{1, \dots, N\}$ to $\{1, \dots, M\}$ which are in one to one mapping with strictly increasing functions from $\{1, \dots, N\}$ to $\{1, \dots, M + N - 1\}$. □

Closed queuing network model for static policies The VSS stochastic evaluation model can be represented for a static policy as a closed queueing network with finite capacities and periodic time-varying service rates. An example with 2 stations is schemed in Figure 2.4. This closed queuing network is built as follows.

There is a fixed number of vehicles circulating in the network, hence it is natural to see the system from a vehicle's perspective. Each station $a \in \mathcal{M}$ is represented by a server a.Vehicles are jobs waiting in these queues for users to take them. The time-varying service rate λ^t_a of server a is equal to the average number of users willing to take a vehicle at station a at time t: $\lambda^t_a := \sum_{b\in\mathcal{M}} \lambda^t_{a,b}$.

At time t, a vehicle taken by a user for a trip $(a, b) \in \mathcal{D}$ is represented by a job processed by server a with routing probability $\frac{\lambda^t_{a,b}}{\lambda^t_a}$. Before arriving at the destination station (server) b, the vehicle (job) passes by a transportation state represented by an infinite server $(a-b)$ with rate $\mu^t_{a,b}$. This infinite server represents users traveling in parallel and independently. It can be seen as a single server with a service rate $n_{a,b}\mu^t_{a,b}$ that is proportional to the number of vehicles $n_{a,b}$ in the queue (in transit).

The N vehicles are N jobs. Vehicles are either in a station or in transit: $N = \sum_{a\in\mathcal{M}} n_a + \sum_{(a,b)\in\mathcal{D}} n_{a,b}$ with n_a the number of vehicles in station a.

The parking spot's reservation at destination constrains the capacity $\mathcal{K}_a$ of station a to be shared between the queue capacity of server a and of servers $(b-a)$. In other words, the $\sum_{b\in\mathcal{M}} n_{b,a}$ vehicles in transit towards station a already occupy a parking spot in a in the same way as the n_a vehicles currently in a: $n_a + \sum_{b\in\mathcal{M}} n_{b,a} \leq \mathcal{K}_a$.

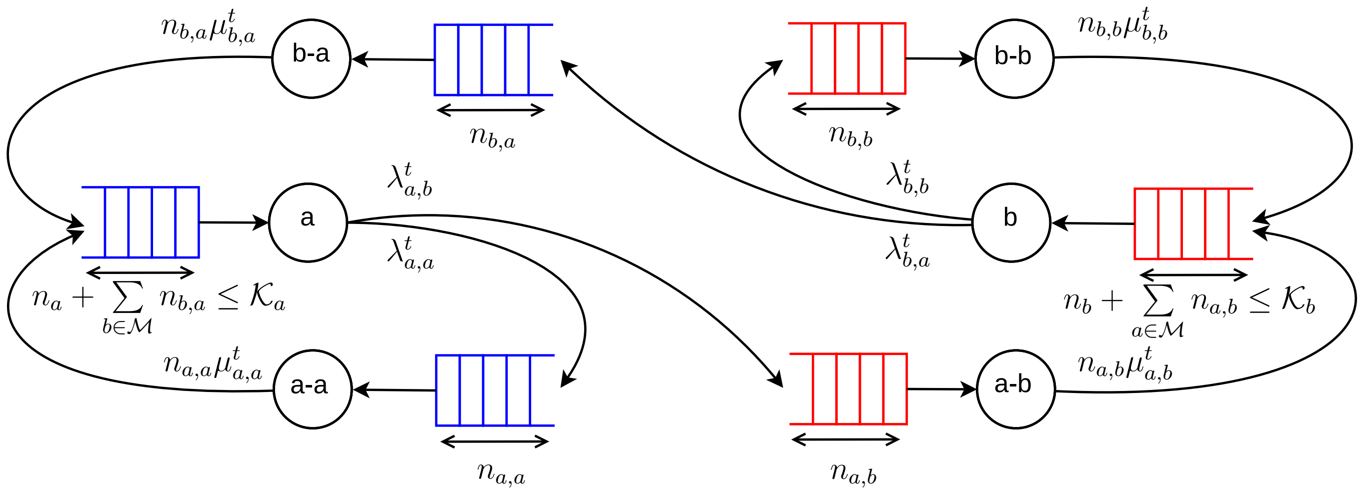


Figure 2.4: VSS stochastic model: A closed queuing network with finite capacities and periodic time-varying rates.

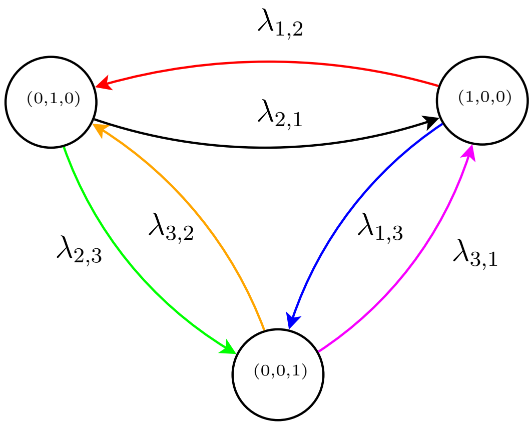
Figure 2.5 considers a city with 3 stations, 2 vehicles, a stationary demand (one time step) and null transportation times. Figure 2.5a represents the demand graph on the *space network*. Each station is represented by a vertex, and a weighted arc represents the rate of the stochastic demand to take a trip between two stations. When there is only 1 vehicle, since there is no transportation times, it is either located in station 1, 2 or 3. Therefore, Figure 2.5a represents also the state graph of the system. For 2 vehicles, as schemed in Figure 2.5b, the system's state graph contains 6 different vehicle distributions (vehicles are not differentiated).

2.2.4 Literature review

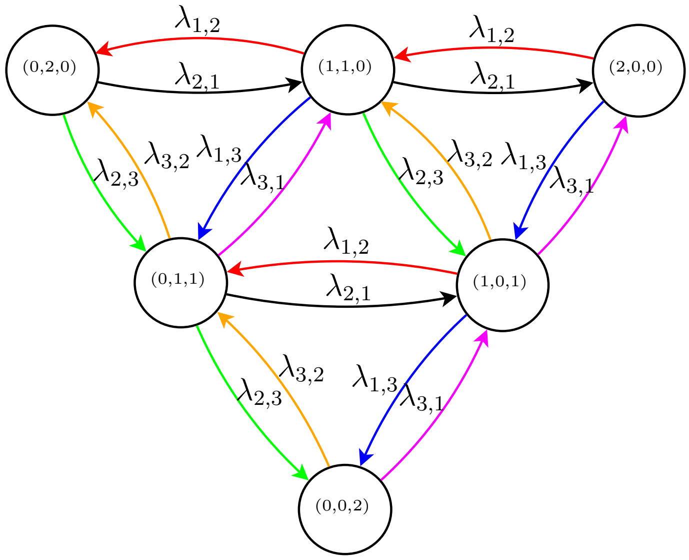
VSS stochastic optimization Simpler forms of the stochastic evaluation model as a closed queuing network are studied in the VSS literature for the fleet sizing problem. George and Xia (2011) consider a VSS with a fixed stationary demand (no pricing) and infinite station capacities. Under these assumptions, they establish a compact form to compute the system performance using the BCMP [4] network theory (Baskett *et al.*, 1975). They solve an optimal fleet sizing problem considering a fixed cost per vehicle and a gain to rent it.

Fricker and Gast (2012) consider toy cities, perfectly balanced, that they call homogeneous. These cities have a unique fixed station capacity ($\mathcal{K}_a = \mathcal{K}$), a stationary demand, a uniform routing matrix ($\lambda_{a,b} = \frac{\lambda}{M}$) and a unique travel time (${\mu_{a,b}}^{-1} = \mu^{-1}$). With a mean field approximation, they obtain asymptotic results

4. It is named after the authors of the paper where the network was first described: Baskett, Chandy, Muntz and Palacios.



(a) Demand graph = State graph for 1 vehicle.



(b) State graph for $N = 2$ vehicles.

Figure 2.5: A city with 3 stations, null transportation times and a stationary demand.

when the number of stations tends to infinity ($M \to \infty$): without regulation systems, the optimal fleet sizing is $\frac{\mathcal{K}}{2} + \frac{\lambda}{\mu}$ vehicles per station which corresponds in half filling each station plus the average number of vehicles in transit toward them ($\frac{\lambda}{\mu}$). Moreover, they show that even with an optimal fleet sizing, each station has still a probability $\frac{1}{\mathcal{K}+1}$ to be empty or full (which is considered a poor performance since these cities are perfectly balanced). In another paper, Fricker *et al.* (2012) extend part of the analytical results to inhomogeneous cities modeled by clusters and they derive some results experimentally.

For homogeneous cities, Fricker and Gast (2012) also study a heuristic using incentives called "the power of two choices" that can be seen as a dynamic pricing. When a user arrives at a station to take a vehicle, he gives randomly two possible destination stations and the system is directing him to the least loaded one. They show that this policy allows to drastically reduce the probability to be empty or full for each station to $2^{-\frac{\mathcal{K}}{2}}$.

None of these models, that are dedicated to VSS, include time-varying demands (service rates), pricing or full heterogeneity.

Queuing network with time-varying rates There is a wide literature on queuing networks and MDPs. We refer to the textbooks of Puterman (1994) or Bertsekas (2005a) to provide the foundation for using MDP for the exact optimization of stationary queueing systems. We now focus our short review on time-varying rates for the average reward criterion.

Queuing networks with time-dependent parameters are called in the literature either dynamic rates queues, time varying rates queues or unstationnary queues. When dealing with Markovian systems, the term inhomogenous MDP is used in opposition to classic homogeneous MDP. Many researchers have extended the MDP framework to develop policies for inhomogenous stochastic models with infinite actions spaces. Yoon and Lewis (2004) consider both pricing and admission controls for a multiserver queue with a periodic arrival and service rate over an infinite time horizon. They use a pointwise stationary approximation (Green and Kolesar, 1991) of the queueing process: an optimization problem is solved over each disjoint time interval where stationarity is assumed.

In his PhD thesis, McMahon (2008) studies how to incorporate time-dependence into the system dynamics of Markovian decision processes. McMahon formulates it as a simple decision process, with exponential state transitions, and solve this decision process using two separate techniques. The first technique solves the value equations directly, and the second utilizes an existing continuous-time MDP solution technique. We finally refer to Liu (2011) PhD thesis that develops deterministic heavy-traffic fluid approximations for many-server stochastic queueing models with time-varying general arrival rates and service-time distributions.

Blocking effect When considering queuing networks with finite capacities, blocking effects arise when a queue is full. Balsamo *et al.* (2000) define various blocking mechanisms. Osorio and Bierlaire (2009) review the existing models and present an analytic queueing network model which preserves the finite capacity of the queues and uses structural parameters to grasp the between-queue correlation.

Blocking mechanisms differ either in the moment the job is considered to be blocked (before or after-service) or in the routing mechanism of blocked jobs. For our VSS queuing network model, we have to distinguish two cases depending on the rental reservation policy:

- If there is no parking spot reservation, when a user tries to return a vehicle at a full station, the system is facing a *Repetitive Service Blocking (RS)*. Two solutions might be considered then: 1) Either the user can choose a new destination station independently from the one he had selected previously, until he finds a free parking spot full. This is known as *RS-RD* (random destination). This is the blocking mechanism considered by Fricker and Gast (2012). 2) Or if he does not modify its destination station, he has to wait for a free parking spot. This is known as *RS-FD* (fixed destination).
- If the user has to reserve a parking spot at destination, the blocking mechanism

is of type *Blocking Before Service* (BBS).

In our closed queuing network model, even if the reservation of parking spots at destination looks like a BBS, the blocking mechanism is somehow special. Indeed, because of transportation times, the blocking constraint links the capacities of several queues: all queues representing the transportation time toward a station a and the queue representing the station a itself; see Section 2.2.3.

2.3 Optimization model – A pricing problem

We now define formally the pricing problem we want to tackle in this thesis.

2.3.1 The VSS stochastic pricing problem

We want to maximize the VSS performance using pricing as leverage. The efficiency of a pricing policy is measured by the VSS stochastic evaluation model. We call this problem the *VSS stochastic pricing* problem.

VSS Stochastic Pricing Problem

- **Instance**:
  - A number N of vehicles;
  - A set $\mathcal{M}$ of stations with capacities $\mathcal{K}_a,\ a \in \mathcal{M}$;
  - A set $\mathcal{T}$ of time steps with duration $\tau^t,\ t \in \mathcal{T}$;
  - For every trip $(a,b) \in \mathcal{M}^2$, at every time step $t \in \mathcal{T}$, the demand set $\Omega^t_{a,b}$ per time unit to take trip (a,b) with transportation time following an exponential distribution with mean $1/\mu^t_{a,b}$:
    [**Discrete Pricing**] $\Omega^t_{a,b} = \{\Lambda^{t,1}_{a,b}, \dots, \Lambda^{t,k}_{a,b}\}$;
    [**Continuous Pricing**] $\Omega^t_{a,b} = [0, \Lambda^t_{a,b}]$.
- **Solution**: :
  [**Dynamic Policy**] A demand $\lambda^t_{a,b}(s) \in \Omega^t_{a,b}$, to take each trip $(a,b) \in \mathcal{D}$ function of the system's state $s \in \mathcal{S}$;
  [**Static Policy**] A tuple $(\lambda,\ k,\ \vec{\mathcal{M}},\ \vec{N})$, where:
  - $\lambda^t_{a,b} \in \Omega^t_{a,b}$ is the demand to take trip $(a,b) \in \mathcal{D}$ at time step $t \in \mathcal{T}$,
  - The connection graph $G(\mathcal{M}, \sum_{t\in\mathcal{T}} \lambda^t)$ defines a set of k strongly connected components $\vec{\mathcal{M}} = \{\mathcal{M}_1, \dots, \mathcal{M}_k\}$,
  - $\vec{N} = (N_1, \dots, N_k)$ is the vehicle distribution over $\vec{\mathcal{M}}$, $(\sum_{i=1}^k N_i = N)$.
- **Measure**: The pricing policy value measured by the stochastic evaluation model on a criteria that can be among others:
  [**Transit Max**] Expected number of trips sold;
  [**Use Max**] Expected vehicle utilization.

In order to consider the problem maximizing the revenue generated, one needs to define a price function $price : \Omega \to \mathbb{R}$ in the input. In this study most results focus on the VSS Stochastic – Continuous Pricing – Static Policy – Transit Maximization problem.

We restrict the study of dynamic policies to the (dominant) class for which the graph spanned by $\{(a,b) \in \mathcal{D},\ s \in \mathcal{S},\ \lambda^s_{a,b} > 0\}$ has only one strongly connected component. Otherwise, the stationary distribution on the state graph is not unique: it depends on the initial state of the system.

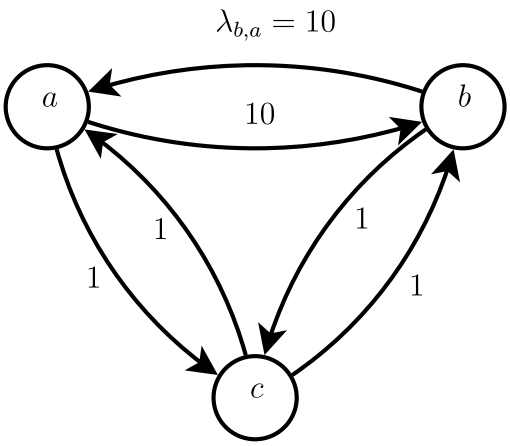
Sometimes optimal static policies need more than one strongly connected components on the station graph. An example is given in Proposition 5 Section 2.3.3.3. The k strongly connected components of the static policy connection graph $G(\mathcal{M}, \sum_{t\in\mathcal{T}} \lambda^t)$ divides the city into k independent VSS, sharing a number N of vehicles. The vehicle distribution has then to be explicitly specified since it impacts the policy performance. For dynamic policies, the vehicle distribution is explicit (defined by the system states for single component policies). That is why for ease of notations the stochastic evaluation model is defined for dynamic policies (any static policy can be represented as a dynamic one).

A static pricing example Figure 2.6 shows an example of 2 static policies in a city with 3 stations, null transportation times and a stationary symmetric demand. Figure 2.6a represents the policy setting all prices to their minimum values, *i.e.* in which the demand is maximal for every trip. For one vehicle this policy sells 8 trips per time unit [5]. Figure 2.6b represents the static policy maximizing the number of trips sold. It consists in closing station c, *i.e.* refusing all trips to station c. For one vehicle, using this policy increases the number of trips sold to 10 per time unit.

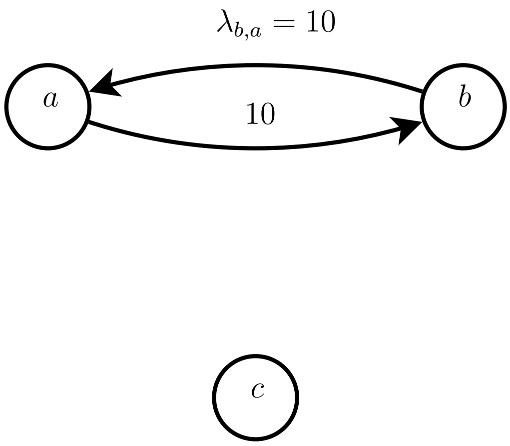
A dynamic pricing example Figure 2.7 schemes an example of an optimal dynamic pricing policy in a city with 2 stations, a stationary demand and null transportation times. Figure 2.7a defines the demand graph with the 3 available prices to take each trip: 3 different couples (demand, price) on each arc. Figure 2.7b represents the optimal dynamic policy for 2 vehicles [6]. A dynamic policy can be represented through the state graph of its induced Markov chain. Notice that the price to take a trip from station 1 to 0 is always equal to 5 (static) and the price to take the opposite trip depends on the system state (dynamic): it is worth 2 if there is no vehicle in station 1 and it is worth 4 otherwise.

5. For this toy instance (of small size), the stochastic evaluation model can be computed exactly with the continuous-time Markov chain formulation.

6. For this size of instance, the optimal dynamic policy can be computed efficiently with the VSS MDP model defined Section 2.3.3.1.

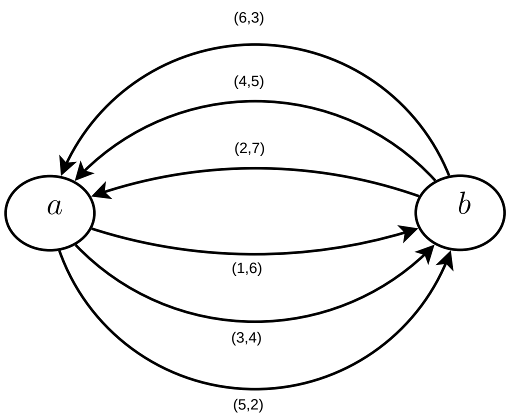


(a) Minimum price policy, 8 trips sold/time unit.

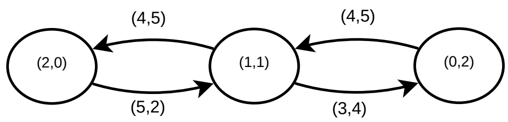


(b) Optimal static policy, 10 trips sold/time unit.

Figure 2.6: Static policy transit optimization, example with 1 vehicle and 3 stations.



(a) Demand graph: (demand, price).



(b) Optimal dynamic policy with gain ≈21.6/time unit, induced Markov chain's state graph for 2 vehicles: (demand, price).

Figure 2.7: Dynamic policy revenue optimization, example with 2 stations, 2 vehicles and 3 discrete prices per trip.

2.3.2 Complexity in a stochastic framework

The previous formal problem definition enables to define *tractability*, *polynomiality* or simply *efficiency* for VSS stochastic pricing optimization. To tackle large scale (real-world) systems, we need solution methods that have computational time polynomial in N, $|\mathcal{M}|$ and $|\mathcal{T}|$. The solutions (pricing policies) produced (output) need also to be of moderate size. Notice that the state graph (of exponential size) representing all possible vehicle distributions (system's states) is not part of the input. The explicit representation of dynamic policies is hence not tractable.

To the best of our knowledge, the problem of measuring exactly the stochastic evaluation model in a polynomial time for a given pricing policy is open. For a simplified model with a stationary demand ($|\mathcal{T}| = 1$) and infinite station capacities measuring exactly the stochastic evaluation model for a static policy is polynomial in M and N. George and Xia (2011) provide a product form formula and algorithms to compute the stochastic evaluation model for a static pricing policy (Remark 1). However, to determine if the static pricing problem [7] belong to NP we need to make some assumptions. All stochastic processes follow exponential distributions, and that exponential distributions are totally defined by their means. The size of the input is then $M^2 \log(\Lambda_{\max})+\log(N)$ assuming that $\Lambda^t_{a,b} \in \mathbb{N}$, $\forall (a, b) \in \mathcal{D}$, $\forall t \in \mathcal{T}$. In practice, $N = O(M)$ therefore we consider that the size of the instance is polynomial in M, N, $\log(\Lambda_{\max})$. If we assume that optimal solutions (a vector $0 \leq \lambda \leq \Lambda$) have an encoding size polynomial in M, N, $\log(\Lambda_{\max})$, the problem [7] is in NP.

The VSS stochastic evaluation model can be estimated efficiently through Monte-Carlo simulations even for very large state spaces. Therefore, we use simulation to compare our proposed pricing policies.

Remark 1 (Product form formula for stationary demand and infinite station capacities)**.** *We recall George and Xia (2011) product form formula based on BCMP queuing network theory (Baskett et al., 1975). For N vehicle, the probability to be in state $s \in \mathcal{S}$ equal:*

$$P\big(s = (n_a :\ a \in \mathcal{M},\ n_{a,b} :\ (a,b) \in \mathcal{D}) \in \mathcal{S}\big) = \frac{1}{G(N)} \prod_{a \in \mathcal{M}} \pi_a^{n_a} \prod_{(a,b)\in\mathcal{D}} \frac{\pi_{a,b}^{n_{a,b}}}{n_{a,b}!}.$$

Where π is the stationary distribution among the continuous-time Markov chain states for a system with only one vehicle: π_a is the stationary probability to have the vehicle in station a and $\pi_{a,b}$ to have it in transit between station a and b. G(N) is

7. We refer here to the associated decision problem: Is there a static pricing policy expecting to sell at least X trips in the stochastic evaluation model?

the normalization constant:

$$G(N) = \sum_{s \in \mathcal{S}} \prod_{a \in \mathcal{M}} \pi_a^{n_a} \prod_{(a,b) \in \mathcal{D}} \frac{\pi_{a,b}^{n_{a,b}}}{n_{a,b}!}.$$

$G(N)$ that can be computed efficiently with the convolution method of Buzen (1973). The availability at station a is equal to $A_a(N) = \pi_a \frac{G(N-1)}{G(N)}$. And finally, the expected number of trips sold by the system can be computed as follows:

$$\sum_{(a,b) \in \mathcal{D}} A_a \lambda_{a,b} = \frac{G(N-1)}{G(N)} \sum_{(a,b) \in \mathcal{D}} \pi_a \lambda_{a,b}.$$

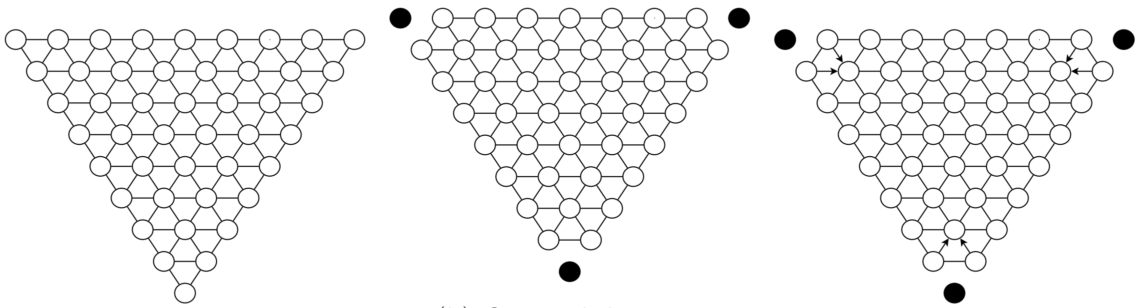
2.3.3 Toward computing optimal policies

Since a straightforward approach (MDP) cannot tackle large scale (real-world) systems, we search for dominant structures that could help the optimization process. We study a simpler model: a stationary demand ($\Lambda^t_{a,b} = \Lambda_{a,b}$), null transportation times and infinite station capacities.

2.3.3.1 Markov Decision Process – The curse of dimensionality

Computing optimal dynamic policies The continuous-time Markov chain formulation of the VSS stochastic evaluation model leads directly to a Markov Decision Process (MDP), named the *VSS MDP model*. This model considers, in each state $s \in \mathcal{S}$, a set $\mathcal{Q}$ of discrete prices for each possible trip. Solving the VSS MDP model computes the optimal dynamic discrete pricing policy.

MDPs are known to be polynomially solvable in the number of states $|\mathcal{S}|$ and actions $|\mathcal{A}|$ available in each state. To solve an MDP, efficient solution methods exist such as value iteration, policy iteration algorithm or linear programming; see Puterman (1994) textbook. In each state $s \in \mathcal{S}$, the VSS MDP model's action space $\mathcal{A}(s)$ is the Cartesian product of the available prices for each trip, *i.e.* $\mathcal{A}(s) = \mathcal{Q}^{|\mathcal{M}|^2}$. The action space size is then exponential in the number of stations. However, to avoid suffering from this explosion, we can model this problem as an action decomposable Markov decision process; it is a contribution of this thesis presented in Appendix A. Thanks to this general framework, based on the event-based dynamic programming (Koole, 1998), the complexity of solving the VSS MDP model becomes polynomial in $|\mathcal{S}|$ and $|\mathcal{Q}||\mathcal{M}|^2$ (that is far less than $|\mathcal{Q}|^{|\mathcal{M}|^2}$). Nevertheless, the VSS MDP model has another problem: the explosion of its state space $\mathcal{S}$ with the number of vehicles and stations. This phenomenon is known as the *curse of dimensionality* (Bellman, 1953).



(a) Policy opening all trips, value 4.8.

(b) Optimal dynamic capacity policy, value ≈ 4.857.

(c) Optimal dynamic policy, value ≈ 4.865.

Figure 2.8: Induced Markov chain of 3 policies evaluated in an homogeneous city with 8 vehicles and 3 stations. Legend: ($\circ$) reachable state; ($\bullet$) unreachable state; ($-$) trip between two states open in both directions; ($\rightarrow$) trip open in only one direction.

2.3.3.2 Structures of optimal dynamic policies

Recall that *Dynamic policies* have prices to take a trip that depend on the state of the system, *i.e.* the vehicle distribution. Unfortunately, even with homogeneous demand ($\Lambda_{a,b} = \Lambda$) optimal dynamic policies seem hard to describe.

Since the number of states is exponential, we would like to restrict to dynamic policies allowing a compact description. *Capacity policies* amount to specifying a virtual station capacity $\mathcal{K}$, and to accept a trip from station a to station b if only if the number of vehicles in b is not exceeding $\mathcal{K}_b$.

We show in the next proposition that capacity policies are suboptimal among dynamic policies for the VSS stochastic pricing optimization problem.

Proposition 2. *Capacities policies are suboptimal among dynamic policies, even in homogeneous cities.*

Proof. Figure 2.8 compares the induced Markov chain (state graph) of three policies in an homogeneous city ($\Lambda = 1$) with 3 stations and 8 vehicles. An edge represents that the trip is open to its maximum in both directions, an arc indicates that it is open only in one way. Figure 2.8a represents the generous policy opening all trips and expects to sells 4.8 trips per time unit. Figure 2.8b represents the optimal dynamic capacity policy and increases the gain to ≈ 4.857. Finally, the optimal dynamic policy is represented in Figure 2.8c, and increases the number of trips sold to ≈ 4.865. $\square$

Figure 2.8 shows that using dynamic pricing policies can increase the number of trips sold by the system even in homogeneous cities (perfectly balanced). Figure 2.9

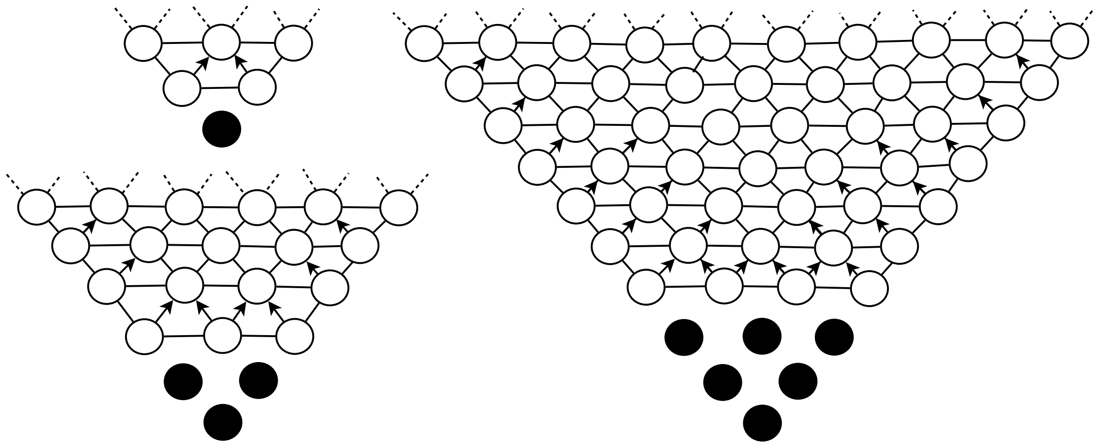


Figure 2.9: "Spikes" of optimal dynamic policies' state graph for an homogeneous city with 3 stations and N=8, 14 or 30 vehicles.

represents the optimal dynamic policies in an homogeneous cities with 3 stations when the number of vehicles increases: from 8 vehicles (as in Figure 2.8b), to 14 and 30 vehicles. Only the "spikes" of the dynamic policies' induced Markov chain are represented since, the solution is invariant under the group S_3 of permutation of the stations. These solutions are the unique optimum [8]. It seems hard to find a compact description of optimal solutions in general.

2.3.3.3 Suboptimal classes of static policies

Generous policies / No regulation When investigating (pricing) policies, the most important practical issue is the trade-off between the simplicity (and in particular, the readability for users) and the performance.

The first practical question might always be whether "unoptimized" policies perform well.

The (static) *generous* policy sets all demands to their maximum value ($\lambda = \Lambda$). To the best of our understanding, the generous policy is the most natural and relevant to compare with in theoretical studies, as long as the objective function is in terms of service quality and not in terms of monetary gain.

In Proposition 3, provides an example in which the number of trips sold by the generous policy can be arbitrarily far from an optimal static policy. It contains a "gravitational" phenomenon, which occurs in particular for bike sharing systems in non-flat cities.

8. The optimal dynamic policy is solved with the VSS (decomposed) MDP model. This model is of exponential size in N and $|\mathcal{M}|$ but still solvable for the size of these 3 instances. The solution uniqueness has been checked greedily solving several decomposed MDPs.

Proposition 3. *The ratio between the number of trips sold by the (static) generous policy ($\lambda = \Lambda$) and the static optimal policy is unbounded.*

Proof. Consider a complete demand graph where all trip maximum demands are equal to 1 except the trips from a special station $z \in \mathcal{M}$ to any other station that are worth L^{-1}: $\Lambda_{a,b} = 1,\ \Lambda_{z,a} = 1,\ \forall a \in \mathcal{M},\ \forall b \in \mathcal{M} \setminus \{z\}$.

For any number of vehicle, when $L \to \infty$ the expected number of trips sold $T(G)$ for the generous policy G tends to 0: The stationary distribution for one vehicle is $\pi_a = \frac{1}{L+M-1},\ \forall a \in \mathcal{M} \setminus \{z\}$ and $\pi_z = \frac{L}{L+M-1}$, hence $\lim_{L\to\infty} \pi_a = 0,\ \forall a \in \mathcal{M} \setminus \{z\}$ and $\pi_z = 1$. Since for all N, the availability vector A satisfies $A = \alpha_N \pi$ for some scalar α_N, we have:

$$\forall N \geq 1, \quad \lim_{L\to\infty} A_a = 0,\ \forall a \in \mathcal{M} \setminus \{z\} \quad \text{and} \quad \lim_{L\to\infty} A_z = 1,$$

hence

$$\forall N \geq 1, \quad T(G) = \sum_{a \in \mathcal{M}} A_a(M-1) + A_z L^{-1}(M-1) \quad \Rightarrow \quad \lim_{L\to\infty} T(G) = 0.$$

On the other hand, the static circulation policy C closing only trips to and from station a has a expected number of trips sold $T(C) > 1$ that is independent of L:

$$\forall L > 0,\ \forall N \geq 1, \quad A_b = \frac{N}{N+M-2},\ \forall b \in \mathcal{M} \setminus \{a\} \quad \text{and} \quad A_a = 0,$$

hence independently of L, and for all $N \geq 1$ and $M \geq 3$

$$T(C) = \sum_{a \in \mathcal{M} \setminus \{z\}} A_a(M-2) = \frac{N(M-1)(M-2)}{N+M-2} \geq 1. \qquad \square$$

Bang bang policies Static policies directly have a compact representation: only one price per trip needs to be set, independently of the system's state.

However, a compact formulation does not directly lead to a polynomial optimization. When considering only two possible prices per trip, a brute force solution method still needs $2^{|\mathcal{M}|^2}$ calls to the stochastic evaluation model. We need to exhibit structures to design efficient algorithms.

With the continuous demand assumption, static policies optimization amounts to setting the user arrival rates λ with $0 \leq \lambda_{a,b} \leq \Lambda_{a,b},\ \forall (a,b) \in \mathcal{D}$. We investigate *bang-bang policies* (all or nothing) that set each trip $(a,b) \in \mathcal{D}$ to be either open ($\lambda_{a,b} = \Lambda_{a,b}$), or closed ($\lambda_{a,b} = 0$). One can wonder if bang-bang policies are dominant for the transit maximization. It is true for dynamic policies: bang-bang dynamic policies optimization can be reduced to a discrete price dynamic policies optimization in which deterministic policies are dominant [9]. Nevertheless, we show that bang-bang policies are not dominant among static policies even (which is more surprising) when the number of vehicles tends to infinity.

Proposition 4. *Bang-bang policies are suboptimal among static policies even when the number of vehicles tends to infinity.*

Proof. Figure 2.10 exhibits a counter example with 4 stations (a, b, c, d) and maximum trip demands $\Lambda_{a,b} = \Lambda_{b,c} = 3$, $\Lambda_{c,d} = \Lambda_{d,a} = \Lambda_{c,a} = 2$, all others are equal to 0. There are only 2 bang-bang static policies λ defining a strongly connected demand graph: $\lambda_{i,j} = \Lambda_{i,j}$, $(i, j) \neq (c, a)$ and either $\lambda_{c,a} = 0$ or $\lambda_{c,a} = 2$. When the number of vehicles tends to infinity, the availability of a vehicle at station a equals $\frac{\pi_a}{\max_{b \in \mathcal{M}} \pi_b}$, where π is the stationary distribution for one vehicle (George and Xia, 2011). For the $\lambda_{c,a} = 0$ policy, we have $\pi_a = \pi_b = \frac{2}{10}$ and $\pi_c = \pi_d = \frac{3}{10} = \pi_{\max}$, so the expected transit when $N \to \infty$ is worth $\frac{\pi_a}{\pi_{\max}}(3 + 3) + \frac{\pi_c}{\pi_{\max}}(2 + 2) = 8$. For the $\lambda_{c,a} = 2$, policy we have $\pi_a = \pi_b = \frac{4}{14}$ and $\pi_c = \pi_d = \frac{3}{14}$, so the expected transit when $N \to \infty$ is worth 10.5 which is thus the optimal bang-bang static policy. Yet, for the non bang-bang policy with $\lambda_{c,a} = 1$ and still $\lambda_{i,j} = \Lambda_{i,j}$, $(i, j) \neq (c, a)$, we have $\pi_a = \pi_b = \pi_c = \pi_d = \frac{1}{4}$, so the expected transit when $N \to \infty$ is worth $11 > 10.5$. Hence, bang-bang policies are suboptimal even when the number of vehicles tends to infinity. □

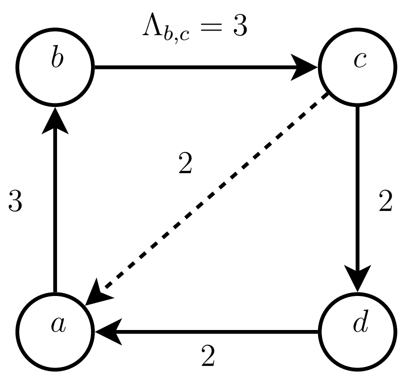


Figure 2.10: Bang-bang policies are suboptimal even when the number of vehicles tends to infinity.

Single component policies One may wonder whether it is useful to have a policy dividing the city. Notice that when considering static pricing policies with more than one strongly connected component, one should explicitly consider the vehicle

9. Classic MDP results (Puterman, 1994).

distribution among these components. In fact, dividing the city sometimes lead to better performances: It is a leverage to prevent the system from being in unprofitable (unbalanced) states.

Proposition 5. *Static policies with one single strongly connected component are suboptimal among static policies.*

Proof. An example is schemed Figure 2.11 with 4 stations and a symmetric demand matrix. For two vehicles, the optimal static policies in this case is to close the trips (b, c) and (c, b) and open all other trips to their maximum value, *i.e.* $\lambda = \Lambda$ except $\lambda_{b,c} = \lambda_{c,b} = 0$. The demand graph of this policy has two strongly connected components. The optimal vehicle distribution is to put one vehicle on each of them. With such distribution it expects to sell 200 trips per time unit. The optimal static policy with a single strongly connected component opens all trips to their maximum value, $\lambda = \Lambda$. It expects to sell 160.8 trips per time unit. □

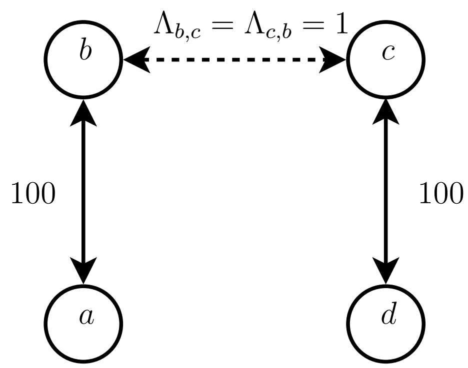


Figure 2.11: Static policies with a single strongly connected component are suboptimal.

2.4 Conclusion

We have presented a stochastic model to tackle the pricing optimization problem in vehicle sharing systems. This model simplifies the real-life problem, though it intends to keep important characteristics such as time-varying demands, station capacities and the reservation of parking spots at destination. In our study, we focus on the transit optimization and therefore do not consider prices explicitly. Hence, we speak about pricing policies but they amount to considering incentive policies or simply policies regulating demand.

We proposed a formal definition for the VSS stochastic pricing problem. Although this formulation is compact and relatively simple, solving in general this problem seems hard. We showed that even an exact measure of the VSS stochastic evaluation model is intractable for real size systems. We discussed notions of complexity in this stochastic framework. It allowed us to specify a frame in our search of tractable solution methods for the VSS stochastic pricing problem.

Chapter 3

Scenario-based approach

An approximate answer to the right question is worth far more than a precise answer to the wrong one.

John Tukey (1915–2000)

Chapter abstract

A direct solution method is intractable to solve the VSS stochastic pricing problem (defined Chapter 2) for the size of systems we want to tackle. We therefore discuss a scenario-based approach, *i.e.* off-line deterministic optimization problems on a given stochastic realization (*scenario*). This deterministic model could be used to provide heuristics and bounds for on-line stochastic optimization. This approach raises a new constraint the *First Come First Served constrained flow* (FCFS flow). We derive three problems based on FCFS flows: a design problem, optimizing station capacities, and two operational problems setting static prices. We show that they are all APX-Hard. We study the upper bound given by the classical Max Flow problem and prove its poor worst case ratio.

Keywords: Scenario-based approach; Pricing; Queuing network; Complexity & approximation; Revenue Management; Graph vertex pricing.

Résumé du chapitre

Nous voulons résoudre le problème stochastique de tarification dans les systèmes de véhicules en libre service présenté Chapitre 2. Une résolution directe est intractable pour la taille de système que l'on veut considérer. Nous étudions donc une approche par scénario, *i.e.* une optimisation déterministe hors ligne sur une réalisation d'un processus stochastique (*un scénario*). Ce modèle déterministe peut être utilisé pour fournir des heuristiques et des bornes sur le problème d'optimisation en ligne. Cette approche soulève une nouvelle contrainte le *flot premier arrivé premier servi*. Nous présentons trois problèmes basés sur cette contrainte : un problème stratégique, l'optimisation de la taille des stations, et deux problèmes opérationnels calculant des politiques tarifaires statiques. Nous montrons qu'ils sont tous trois APX-hard. Nous étudions une borne supérieure donnée par le Flot Max et prouvons sa faible performance dans le pire cas. Enfin nous montrons que le Flot Max peut donner un algorithme d'approximation de faible performance mais intéressant d'un point de vue complexité.

Mots clés : Politiques tarifaires ; Approche par scénario ; Réseau de files d'attentes ; Complexité & approximation ; Revenue Management ; Graph vertex pricing.

Contents

This chapter is based on the article "Vehicle Sharing System Optimization: Scenario-based approach" (Waserhole *et al.*, 2013b) submitted to The European Journal of Operational Research.

3.1 Introduction

In practice there is a lot of uncertainty in VSS dynamic. Dealing with human behavior, variability of user arrivals and transportation times has an important influence. In this context, stochastic optimization seems the most relevant approach to cope with randomness. In Chapter 2 we propose a stochastic model for the VSS stochastic pricing problem. For this model, a naive direct optimization with a Markov Decision Process computing the best dynamic (state dependent) policy is intractable: it can't even scale up for systems in the order of 7 stations. This problem is known as the curse of dimensionality; the number of states of the induced Markov chain is exponential and hence exact solution techniques are not applicable. In this chapter, we study a deterministic approximation, the scenario-based approach, for the VSS stochastic pricing problem defined Chapter 2.

When dealing with stochastic problems, it is classic and natural to consider deterministic approximations. The scenario-based approach amounts to optimizing a posteriori the system, considering that all trip requests (a *scenario*) are available at the beginning of the time horizon. Morency *et al.* (2011) show that, in Montreal's BSS Bixi (2009), 68% of the trips were made by "members" and that their frequencies of use are quite stable along the week. For this context, considering deterministic requests might be a good approximation.

This approach offers two main advantages: On the one hand, the off-line deterministic optimization solution gives a bound for on-line stochastic optimization on a given instance; On the other hand, solving efficiently the deterministic problem on a scenario is the first step toward robust optimization methods (Bertsimas *et al.*, 2011b), at least for models describing uncertainty by sets of scenarii.

Although this paper deals with VSS optimization, the theoretical problem addressed is the optimal control of closed queuing networks with general service time and arrival rate distributions. Therefore, our results can be applied to a wider class of queuing network problems to conduce performance analysis (Bertsimas *et al.*, 2011a) or to estimate the relevancy of robust optimization.

The remaining of this chapter is structured as follows: In Section 3.2, we describe a new type of constraint implied by the VSS scenario-based approach: the *First Come First Served constrained flow* (FCFS flow). In Section 3.3, we define a station capacity problem based on the FCFS flow that is shown APX-hard. In Section 3.4, we define two pricing problems based also on this constraint that are both shown APX-hard: 1) The trip pricing problem that decides a price for taking each trip and 2) The station pricing problem that decides for each station the price to take and return a vehicle. In Section 3.5, we study a bound and an approximation algorithm for FCFS flow pricing problems based on the Max Flow algorithm. Finally in Section 3.6, we study the complexity of a different deterministic problem that does not involve any FCFS flow rule: the optimization of trip reservation in advance.

3.2 First Come First Served constrained flows

Vehicle moves can be modeled as a new type of constrained flow over a time and space network: the *First Come First Served constrained flow* (FCFS flow). Even if not explicitly specified nor named, this constraint is implicitly present in some continuous time models. For instance, it arises naturally in many applications such as in the fluid approximation of a Markov Decision Process (Maglaras, 2006; Waserhole and Jost, 2013b). However, to the best of our knowledge, the FCFS constrained flow is usually implicitly respected in continuous-time models and it has not been studied nor mentioned yet in discrete-time problems.

In the sequels, in order to remain in the lexical field of VSS, we speak about a flow of vehicles transiting among stations thanks to users. Nevertheless, in the more general context of queuing networks, it can be seen as a flow of clients moving along servers.

3.2.1 FCFS flow in time and space network

We consider a system of N vehicles transiting among a set S of stations with infinite capacities. The time horizon is $H = [0, T]$ and at time 0 the distribution of the vehicles among the stations is known. A trip request $r \in R$ asks for a vehicle from an origin station s^r_o at time t^r_o to a destination station s^r_d at time t^r_d. The vehicles move like an automatic flow, *i.e.* no decision can influence the moves. As time goes on, the vehicles transit between stations by accepting the first spatio-temporal trip requests they meet, hence applying the FCFS rule.

We can build a time and space network to follow the evolution of the process. From the beginning of the horizon, we increase the time until an event (trip request or vehicle arrival) occurs. We assume that no two events occur exactly at the same instant. At time t, the trip request $r = (s_o^r, t_o^r = t, s_d^r, t_d^r) \in R$ is accepted if and only if there is a vehicle available at station s_o^r at this time. If trip request r is accepted, a vehicle is removed from station s_o^r and it will be available again at time t_d^r at station s_d^r. If the trip is rejected, nothing happens.

We call this process *First Come First Served constrained flow (FCFS flow)*. Figure 3.1 schemes an example of a FCFS flow with 3 stations, 12 requests and 2 vehicles, one available at station a and the other one available at station c at the beginning of the horizon. In this *scenario*, with 2 vehicles, only 5 trip requests among 12 are served.

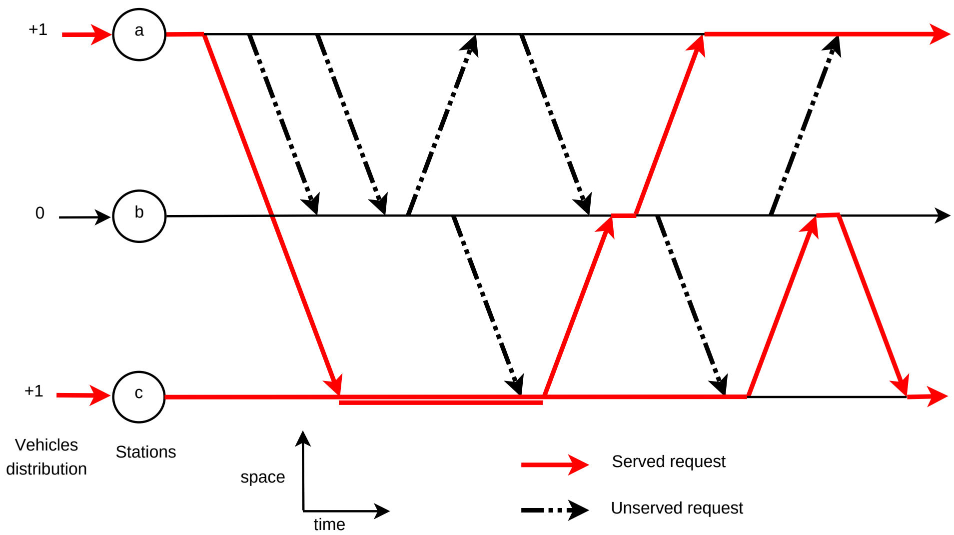


Figure 3.1: An example of a FCFS flow with 2 vehicles and 5 trip requests served.

3.2.2 Station capacity

If we consider now that station $s \in S$ has a capacity $\mathcal{K}_s$, blocking effect issues arise when a station is full. In theory, overbooking or client waiting time penalty might be interesting to study. However in practice, in car VSS, users have the possibility to reserve a parking spot at destination to be sure to be able to retrieve the vehicle. Therefore, in order to avoid blocking effects, we assume that every trip is taken with a parking spot booked at destination. Formally, with station capacities and parking spot reservation, a trip request $r = (s_o^r, t_o^r = t, s_d^r, t_d^r) \in R$ is accepted if and only if there is a vehicle available at station s_o^r at time t and a parking spot available at station s_d^r also at time t.

3.2.3 Priced FCFS flows

We now enhance the system with prices. A price $p^r_{\max}$ is associated to request $r \in R$. This price is the maximum amount the user is willing to pay for taking the trip. The system proposes a fixed price $p_{a,b}$ for each trip $(a, b) \in S^2$. The set of requests that can be served is now reduced to $R_p = \{r \in R \ : \ p^r_{\max} \geq p_{s^r_o, s^r_d}\}$, namely the requests that can afford the price proposed by the system. If request r is accepted, it generates then a gain $p_{s^r_o, s^r_d}$. We call this process *priced FCFS flow.*

Figure 3.2 schemes an example of the run of such a process with 3 stations and 1 vehicle. The graph on the left represents the space network that indicates the prices proposed by the system. For this example, with 1 vehicle available at station a at the beginning of the horizon, 10 trip requests among the 12 can afford the asked price and 6 requests are served for a gain of 49.

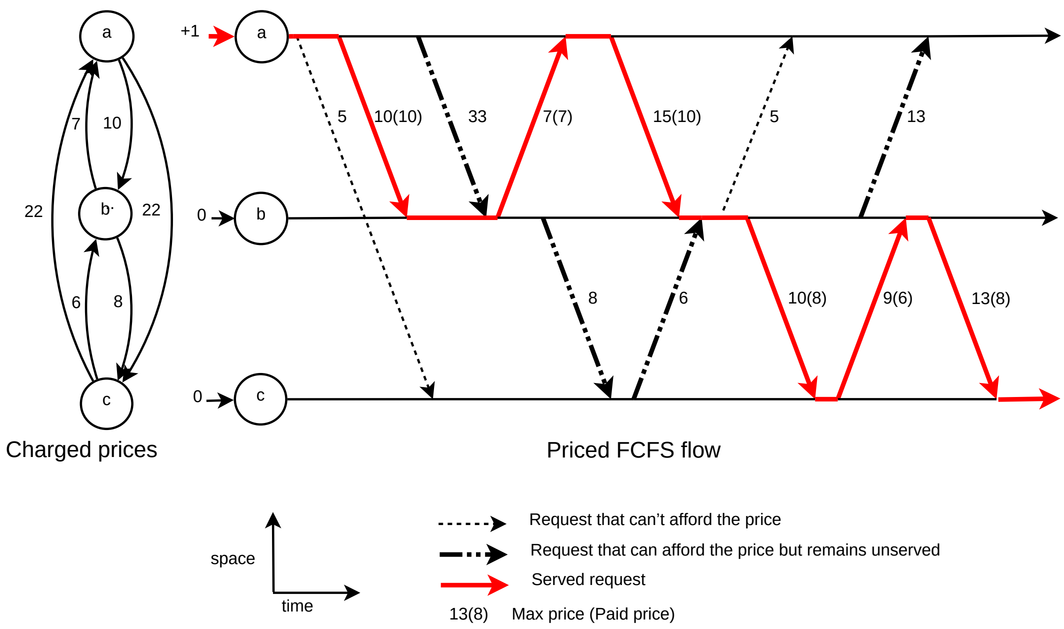


Figure 3.2: Priced FCFS flow with one vehicle and gain 49.

Formally, with station capacities and parking spot reservation at destination, a trip request $r = (s^r_o, t^r_o = t, s^r_d, t^r_d, p^r_{\max}) \in R$ is accepted if and only if there is a vehicle available at station s^r_o at time t, a parking spot available at station s^r_d also at time t and the user is willing to pay the proposed price, *i.e.* $p^r_{\max} \geq p_{s^r_o, s^r_d}$.

Remark 2. *The gain generated by a FCFS flow can be evaluated in linear time. Hence the decision versions of the optimization problems considered in the following are in NP.*

3.3 Station capacity problem

In this section we study the complexity of a tactical problem: setting a capacity for each station such that the number of trips sold in a FCFS flow for a set of trip requests is maximized.

Intuitively, without any additional constraints, one would like to set all station capacities to the number of vehicles, *i.e.* $\forall s \in S$, $\mathcal{K}_s = N$. However, it might be interesting to set smaller values for $\mathcal{K}$ in order to control the location of vehicles in a system with tide phenomenons for instance. Station capacities are then used as a balancing tool. Figure 3.3 schemes an example of station capacity optimization. For this instance, the optimal capacity for station a is $\mathcal{K}_a = N/2$ while station b and c have a capacity $\geq N$. With this sizing, $N/2$ vehicles are taken by half of the trip requests from station b to station a at price 1 until station a is full. Then the remaining vehicles wait in station b before serving all trip requests going to station c at price 2. This policy generates the optimal final profit of $3N/2$ whereas setting all station capacities to N would lead to a profit of N.

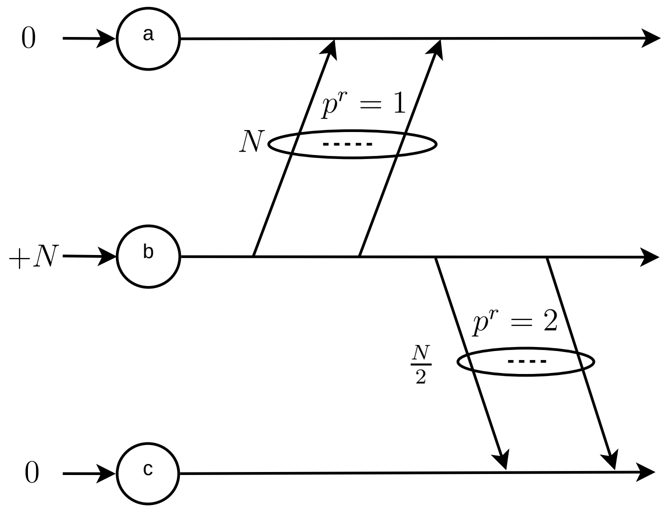


Figure 3.3: Example where proper station capacities increase the number of trips sold. Here setting $\mathcal{K}_a = N/2$ and $\mathcal{K}_b = K_c \geq N$ gives the optimal revenue of $3N/2$.

We now formalize the problem and derive some complexity results.

Max FCFS Flow Station Capacities

- **Instance**: A set of stations S, a number N of vehicles with their distribution among the stations at the beginning of the horizon, a set of trip requests $r \in R$ to go from an original station s_o^r at time t_o^r to a destination station s_d^r at time t_d^r for a price p^r.

- **Solution**: A function $\mathcal{K} : S \to \mathbb{N}^+$ defining the capacity of each station.
- **Measure**: The gain generated by the FCFS flow with station capacities $\mathcal{K}$.

Theorem 1. Max FCFS Flow Station Capacities *problem is NP-hard even with one vehicle and unitary maximum prices.*

Proof. We reduce any instance (with n variables and m clauses) of the NP-complete problem 3-SAT (Garey and Johnson, 1979) to an instance of Max FCFS Flow Station Pricing with one vehicle. Figure 3.4 schemes an example of such a reduction with two clauses. To each variable $\dot{v}$ of a 3-SAT instance, we associate 3 stations $\dot{v}$, v and $\bar{v}$ corresponding to the values unassigned, true and false. We define also two special stations res and tmp. The unique vehicle is located at station res at the beginning of the horizon.

All requests have unitary maximum prices and they are built as follows: Each of the m clauses is taken iteratively. The first clause, let's say $a \vee \bar{b} \vee c$, contains variables $\dot{a}$, $\dot{b}$ and $\dot{c}$. At time 1, there is a request from station res to the station representing the first variable $\dot{a}$. At time 2, the assignment of variable $\dot{a}$ is modeled with two requests in this specific order: from stations $\dot{a}$ to a and then from $\dot{a}$ to $\bar{a}$. At time 3, there is a request from the station representing the literal a contained in the clause to station res. Then, there is another request from station $\bar{a}$, representing the complement of the literal contained in the clause, to the station representing the next variable $\dot{b}$. At time 4, there are two successive requests, from station res to tmp and then from station tmp to res. At time 5, to treat the next variable $\dot{b}$, there is the same series of requests as in times 2, 3 and 4 but adapted to the current variable $\dot{b}$. At time 6, for the last variable of the clause $\dot{c}$, again, there is the same series of requests as in times 2, 3 and 4 adapted to this variable. However, this time, the last request returns to station res. This construction is then repeated for the next clauses.

For a given clause, in the time frame of its associated demands, the longest weighted path has a length and a gain equal to 9. There are 3 different longest weighted paths but all of them are starting and ending at station res. The maximum possible gain is then 9 and it is reached if and only if the assignment of variables satisfies the current clause. Finally there exists a Max FCFS Flow Station Capacities solution on this instance with gain $9m$ if and only if the corresponding 3-SAT instance is satisfiable. Indeed, any 3-SAT satisfiable solution with variable $\dot{v}^{3-SAT}$ can be transform into a Max FCFS Flow Station Capacities solution on the corresponding instance with gain $9m$ thanks to the following mapping: If $\dot{v}^{3-SAT} = true$ then station v is open, otherwise station v is closed and station $\bar{v}$ is open. For the opposite direction: If station v is open then $\dot{v}^{3-SAT} = true$, otherwise

$\dot{v}^{3-SAT} = false$. Remark that one can open at the same time in the MAX FCFS FLOW STATION CAPACITIES instance a station a and a station $\bar{a}$. However it is not a problem since for only one vehicle, when the capacity of station a is equal to 1, the capacity of station $\bar{a}$ is not relevant because there will not be any flow going to station $\bar{a}$. Indeed in our construction, there is always a request to go from station $\dot{a}$ to station a before a request going from station $\dot{a}$ to station $\bar{a}$. ☐

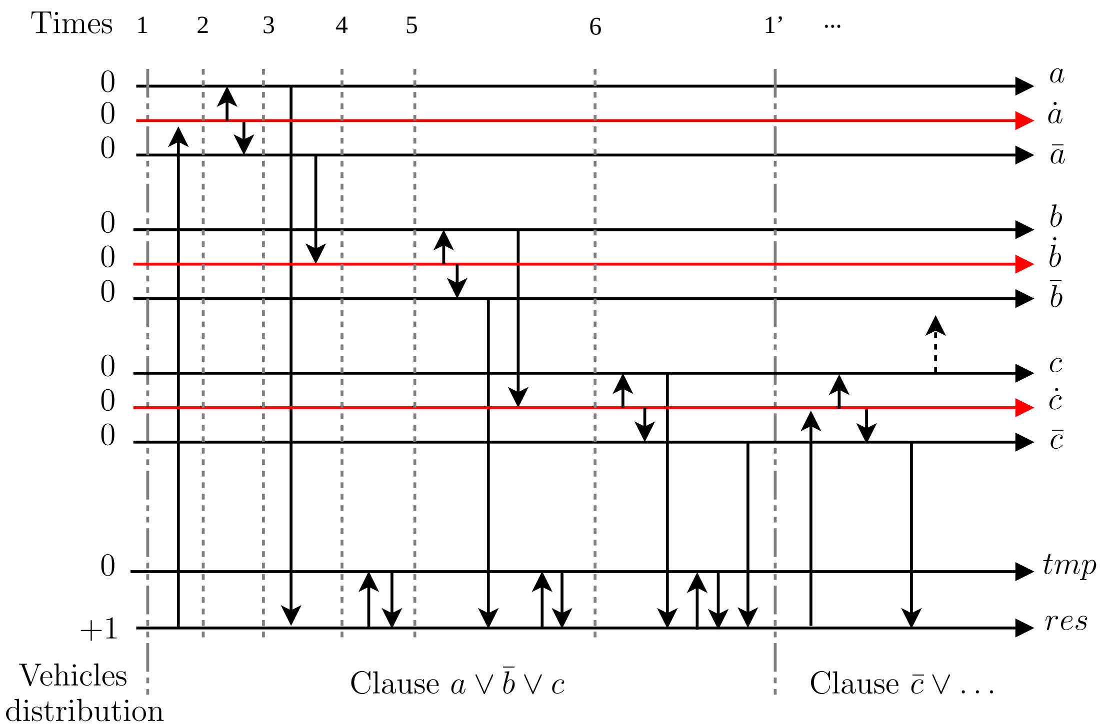


Figure 3.4: Reduction of 3-SAT to FCFS FLOW STATION CAPACITIES. Example with clauses $(a \vee \bar{b} \vee c) \wedge (\bar{c} \vee \ldots)$.

Corollary 1. MAX FCFS FLOW STATION CAPACITIES *problem is APX-hard and not approximable within 39/40 even with one vehicle.*

Proof. MAX-3-SAT is the optimization problem associated to 3-SAT: given a 3-CNF formula, find an assignment that satisfies the largest number of clauses. We use the same construction as in the proof of Theorem 1 to reduce any MAX-3-SAT instance to a MAX FCFS FLOW STATION CAPACITIES instance with one vehicle. In the MAX FCFS FLOW STATION CAPACITIES instance, if a clause is not satisfied, the longest path is 7 and can always be obtained disregarding the variable assignment. Therefore, MAX-3-SAT has a solution with k clauses satisfied if and only if the MAX FCFS FLOW STATION CAPACITIES instance has a solution with gain $9k + 7(m - k) = 2k + 7m$.

Suppose that there exists an algorithm $\mathcal{A}$ for the Max FCFS Flow Station Capacities problem giving a solution of value $F^{\mathcal{A}}$ with approximation ratio $\alpha \in [0,1]$ from the optimal value F^*, *i.e.* $\frac{F^{\mathcal{A}}}{F^*} \leq \alpha$. For the instance built from MAX-3-SAT we have $F^{\mathcal{A}} = 2k^{\mathcal{A}} + 7m$ and $F^* = 2k^* + 7m$. Then:

$$\frac{2k^{\mathcal{A}} + 7m}{2k^* + 7m} \geq \alpha \quad \Leftrightarrow \quad 2k^{\mathcal{A}} \geq 2\alpha k^* + 7m(\alpha - 1). \tag{3.1}$$

A 3-SAT instance always admits a variable assignment satisfying at least 7/8 of the clauses (Karloff and Zwick, 1997), *i.e.* $k^* \geq \frac{7}{8}m$. Since $1 - \alpha \geq 0$ we have $m(\alpha - 1) \geq \frac{8}{7}k^*(\alpha - 1)$. Together with (3.1), it implies:

$$\frac{k^{\mathcal{A}}}{k^*} \geq 5\alpha - 4. \tag{3.2}$$

MAX-3-SAT is not approximable within 7/8 unless P=NP (Karloff and Zwick, 1997), *i.e.* $\frac{k^{\mathcal{A}}}{k^*} \leq \frac{7}{8}$. Together with (3.2), we have:

$$5\alpha - 4 \leq \frac{7}{8} \Leftrightarrow \alpha \leq \frac{39}{40}.$$

Hence Max FCFS Flow Station Capacities is not approximable within 39/40 unless P=NP. □

3.4 Pricing problems

In Section 3.3 we discussed the complexity of a tactical problem, the station capacity design. We now study the complexity of an operational problem: the system management optimization through price leverage. We are searching for pricing policies maximizing the gain of the induced priced FCFS flow.

This investigation leads to the definition of two optimization problems which are both shown APX-Hard: the trip pricing problem which sets a price for each origin-destination pair independently and the station pricing problem which sets, for each station, a price for taking and a price for returning a vehicle. Note that the complexity results can be extended to *time dependent prices* (as long as prices remain constant on some time intervals). Time dependent prices allow to have different prices in the morning, middle of the day and evening in order to control the tide phenomenon for instance.

3.4.1 FCFS Flow Trip Pricing problem

We define the Max FCFS Flow Trip Pricing Problem which consists in setting a price for each trip in order to maximize the gain of the induced priced FCFS flow.

Max FCFS Flow Trip Pricing

- **Instance**: A set of stations S with capacities $\mathcal{K}_s$ for $s \in S$, a number N of vehicles with their distribution among the stations at the beginning of the horizon, a set $R = \{(s_o^r, t_o^r, s_d^r, t_d^r, p_{\max}^r),\ r \in R\}$ of trip requests.
- **Solution**: The prices $p:\ S^2 \to \mathbb{R}$ to take a trip.
- **Measure**: The gain generated by the priced FCFS flow with prices p.

To study Max FCFS Flow Trip Pricing complexity, we extend the approach used for Max FCFS Flow Station Capacities in the previous section.

Theorem 2. Max FCFS Flow Trip Pricing *problem is APX-hard and not approximable within 39/40, even with one vehicle and unitary maximum prices.*

Proof. We reduce a MAX-3-SAT instance to a Max FCFS Flow Trip Pricing instance with one vehicle with the same reduction as in the proof of Theorem 1. Moreover, we consider that all requests have a unitary maximum price: *i.e.* $p_{\max}^r = 1$, $\forall r \in R$. There is a bijection between an optimal MAX-3-SAT solution and an optimal Max FCFS Flow Trip Pricing solution for this instance with the following relation: trips to station a are closed, *i.e.* $p_{\dot{a},a} = \infty$, and trips to station $\bar{a}$ are open, *i.e.* $p_{\dot{a},\bar{a}} = 1$, if and only if variable a is false. Finally, the proof of Corollary 1 can be applied again to show that Max FCFS Flow Trip Pricing is not approximable within 39/40 unless P=NP. $\square$

Remark 3. *If a FCFS flow problem is* hard *even for one vehicle, then it is also* hard *if stations have infinite capacities. Therefore* Max FCFS Flow Trip Pricing *is APX-hard even with infinite capacities.*

3.4.2 FCFS Flow Station Pricing problem

We now consider another way to set the prices $p(a,b)$ to take a trip $(a,b) \in S^2$. It is an aggregation (addition) of a price $p_t(a)$ to take a vehicle in station a and $p_r(b)$ to return it in station b: $p(a,b) = p_t(a) + p_r(b)$. We name it the Max FCFS Flow Station Pricing Problem.

This type of pricing has an interest in a context where users have several possibilities for origin/destination stations. It can help them to figure out quickly the different options they have to take a trip, using for example a price heated maps as in Papanikolaou (2011): stations are colored depending on their prices, for instance from yellow for cheap to red for expensive.

We study the complexity of Max FCFS Flow Station Pricing. Without loss of generality, we consider that prices are independent from the distance/time the vehicle is used. We show that this problem is already *hard* in the single choice context, *i.e.* users only have one possibility for the origin/destination pair.

Max FCFS Flow Station Pricing

- **Instance**: A set of stations S with capacities $\mathcal{K}_s$ for $s \in S$, a number N of vehicles with their distribution among the stations at the beginning of the horizon, a set $R = \{(s_o^r, t_o^r, s_d^r, t_d^r, p_{\max}^r),\ r \in R\}$ of trip requests.
- **Solution**: Prices to take and return a vehicle at a station, p_t and p_r: $S \to \mathbb{R}$.
- **Measure**: The generated gain induced by the priced FCFS flow with prices $p_{a,b} = p_t(a) + p_r(b)$.

Theorem 3. Max FCFS Flow Station Pricing *is APX-HARD and not approximable within 39/40 even with one vehicle or infinite station capacities.*

Proof. We reduce a Max FCFS Flow Trip Pricing instance (Trip-Inst) to a Max FCFS Flow Station Pricing instance (Station-Inst).

Station-Inst is composed with the same set of stations as Trip-Inst plus 2 new stations, ab^1 and ab^2, for each possible trip (a, b). For each trip request $r = (s_o^r = a, t_o^r, s_d^r = b, t_d^r, p_{\max}^r)$ of Trip-Inst , Station-Inst has 3 trip requests: $(a, t_o^r, ab^1, t_o^r + \epsilon, 0)$, $(ab^1, t_o^r + 2\epsilon, ab^2, t_o^r + 3\epsilon, p_{\max}^r)$ and $(ab^2, t_o^r + 4\epsilon, b, t_d^r, 0)$, with ϵ such that $0 < 4\epsilon < t_d^r - t_o^r$.

Note that Station-Inst solutions with $p_t(a) = p_r(ab^1) = p_t(ab^2) = p_r(b) = 0,\ \forall a, b \in S$ are dominant. Moreover, there is a transformation respecting the objective value between an optimal Trip-Inst and an optimal Station-Inst with the relation $p_{a,b} = p_t(ab^1) + p_r(ab^2)$ for each possible trip (a, b). Trip-Inst has a solution of gain at least g if and only if Station-Inst has a solution of gain at least g. Theorem 2 proves that Max FCFS Flow Trip Pricing is APX-hard and not approximable within 39/40 even with one vehicle, therefore Max FCFS Flow Station Pricing is also APX-hard with the same ratio. As in Remark 3, it is also APX-hard for infinite station capacities. □

3.4.3 FCFS flow relaxation: Graph Vertex Pricing

In Theorem 3 we showed that Max FCFS Flow Trip Pricing can be reduced to Max FCFS Flow Station Pricing. The opposite reduction doesn't seem trivial. In fact, there is another difficulty in Max FCFS Flow Station Pricing not related to the flow constraint: the quadratic price assignment. We therefore consider subproblems of Max FCFS Flow Station Pricing where we relax the flow constraint: the Max Oriented Graph Vertex Pricing (O-GVP) problem and its unoriented version Max Graph Vertex Pricing (GVP). We prove that they are already both APX-hard.

Let $G(V, A, c)$ be a weighted directed multi-graph. Vertices V represent the stations and arcs $e \in A$ the trip requests with a weight c_e for the maximum affordable prices. The problem is to set two prices to take and return a vehicle, $p_t(a)$ and $p_r(a)$, for each vertex/station $a \in V$ in order to maximize the total gain on the arcs. A gain of $p_t(a) + p_r(b)$ is generated for each arc $(a, b) \in A$ if and only if $p_t(a) + p_r(b) \leq c_{a,b}$. More formally:

Max Oriented Graph Vertex Pricing (O-GVP)

- **Instance**: A weighted directed multi-graph $G(V, A, c)$ with $c : A \to \mathbb{R}$.
- **Solution**: Prices p_t and p_r: $V \to \mathbb{R}$.
- **Measure**: The generated gain:

$$\sum_{\substack{(a,b)\in A \ / \\ p_t(a)+p_r(b)\leq c_{a,b}}} p_t(a) + p_r(b).$$

We extend the previous definition to weighted undirected multi-graph $G(V, E, c)$. We have to set only one price $p(a)$ for each vertex $a \in V$ in order to maximize the total gain on the edges. A gain of $p(a) + p(b)$ is generated for each edge $(a, b) \in E$ if and only if $p(a) + p(b) \leq c_{a,b}$. More formally:

Max Graph Vertex Pricing (GVP)

- **Instance**: A weighted undirected multi-graph $G(V, E, c)$ with $c : E \to \mathbb{R}$.
- **Solution**: Prices p: $V \to \mathbb{R}$.
- **Measure**: The generated gain:

$$\sum_{\substack{(a,b)\in E \ / \\ p(a)+p(b)\leq c_{a,b}}} p(a) + p(b).$$

Problem GVP has already been studied in the literature. It is one of the fundamental special cases of the Single-Minded item Pricing (SMP) problem (Guruswami *et al.*, 2005). Khandekar *et al.* (2009) prove that GVP is APX-hard on bipartite graphs. The best known approximation algorithm, by Balcan and Blum (2006), gives a 4-approximation. We now present a polynomial reduction from GVP to O-GVP to show that the latter is also APX-hard.

Theorem 4. Max Oriented Graph Vertex Pricing *is APX-hard even on bipartite graphs.*

Proof. We reduce a GVP instance to a O-GVP instance. GVP is APX-hard even on bipartite graphs (Khandekar *et al.*, 2009). A bipartite graph $G(V_1, V_2, E)$ can be oriented such that all vertices of V_1 are sources and all vertices of V_2 are sinks. On this oriented graph, O-GVP solves GVP. Hence, O-GVP is APX-hard even on bipartite graph. □

We use the fact that Max Oriented Graph Vertex Pricing is APX-hard to return to our original problem, Max FCFS Flow Station Pricing and to refine its complexity.

Corollary 2. Max FCFS Flow Station Pricing *is APX-hard even with an unlimited number of vehicles, infinite station capacities or requests defining a bipartite graph.*

Proof. Solving an instance of Max FCFS Flow Station Pricing with an unlimited number of vehicles and infinite station capacities is equivalent to solve an instance of O-GVP in which each request is an arc with weight its maximum price. Max Oriented Vertex Pricing is shown NP-hard on bipartite graphs, therefore Max FCFS Flow Station Pricing is APX-hard even with requests defining a bipartite graph. □

Remark 4. *At the beginning of the section we said that the reduction from* Max FCFS Flow Station Pricing *to* Max FCFS Flow Trip Pricing *is not trivial. Actually Corollary 2 is proving that such reduction cannot exist unless P=NP. Indeed, for an unlimited number of vehicles* Max FCFS Flow Trip Pricing *amounts to solving an* Arc Pricing *problem that is solvable by a greedy polynomial algorithm (decomposing the problem for each arc). Therefore since* Max FCFS Flow Station Pricing *is APX-hard even for an unlimited number of vehicles, it cannot be reduced to* Max FCFS Flow Trip Pricing*.*

3.5 Connections to the Max Flow problem

Given that FCFS flow problems presented in the previous sections are APX-hard, bounds or approximation algorithms might be of interest. A "classic" flow is a relaxation of the first come first served flow evaluation. One of the most famous optimization problem on classic flows is Max Flow which is polynomially solvable. Max Flow gives an Upper Bound (UB) on many FCFS optimization problems such as Max FCFS Flow Station Capacities or Max FCFS Flow Trip/Station Pricing.

In practice, we observe by simulation in Chapter 6 that the ratio between the Max Flow and FCFS flow problems is roughly within a factor 2. In Section 3.5.1, we show that the theoretical guaranty (worst case) of this UB is extremely poor. In Section 3.5.2, we refine on the Max Flow UB through an approximation algorithm for the FCFS Flow 0/1 Trip Pricing, *i.e.* the FCFS Flow Trip Pricing with unitary maximum prices.

3.5.1 Max Flow upper bounds for FCFS flow problems

Max Flow Classic flows don't take into account reservation of parking spots at the destination station. Therefore Max Flow gives an UB that can be arbitrarily far from any FCFS flow. Figure 3.5 schemes an example with 2 stations of unitary capacity and 2 vehicles with q crossed demands. In this example, Max Flow is able to serve all q requests while any FCFS flow with reservation can't serve any.

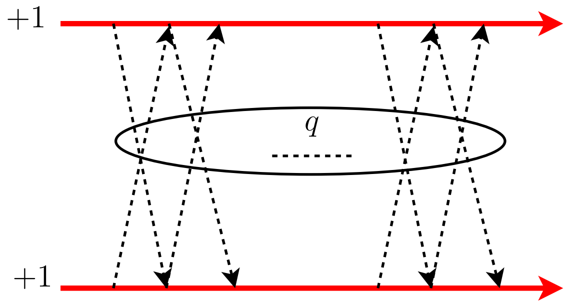


Figure 3.5: Max Flow UB can be arbitrarily far from any FCFS flow since it doesn't consider parking spot reservation.

Max Flow With Reservation Assuming that no two requests arrive at the same time, we can add constraints to the Max Flow classic linear program to respect parking spot reservations. As schemed in Figure 3.6, it amounts to considering requests with null transportation time, respecting station capacities, and then a time where the vehicle is unavailable at the station. The case represented Figure 3.5 is then avoided. We call this problem Max Flow With Reservation (Max Flow WR). Max Flow WR remains polynomial. However, solving it with a classic linear programming solver is much slower than Max Flow because classic flow algorithms do not apply anymore (see Section 6.5.2 page 138).

Max Flow WR can again be arbitrarily far from any FCFS flow. Figure 3.7 schemes it on an example with 2 stations, Lower (L) and Upper (U), 1 vehicle available at L at the beginning of the horizon and trip requests with unitary maximum prices. The first request goes from L to U and takes the entire horizon to reach the

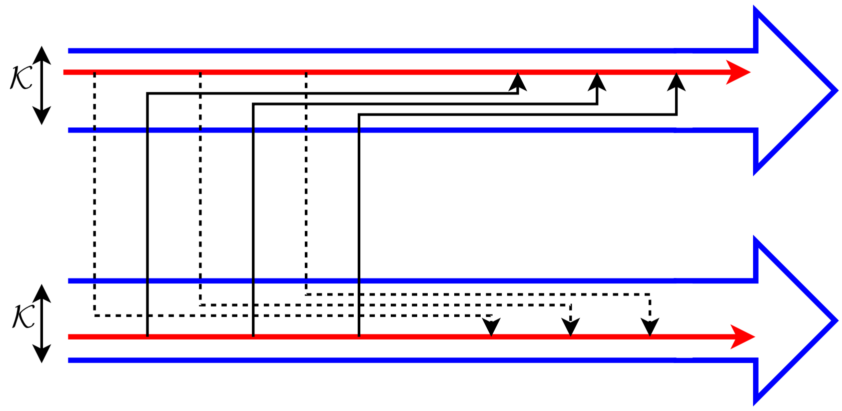


Figure 3.6: A Max Flow With Reservation, 2 stations of capacity $\mathcal{K}$.

station U. Then there are q successive trip requests from L to U and from U to L. In this instance, Max Flow WR is able to serve q requests, rejecting only the first long one, while any FCFS flow can't serve more than one request, the first one.

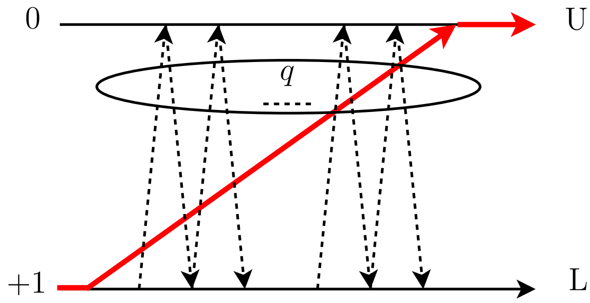


Figure 3.7: The ratio between Max Flow With Reservation and any FCFS flow can be greater unbounded for any $M \geq 3$ and $N \geq 1$.

Max Flow WR for non-crossing requests The previous example used *crossing requests* for the same trip: *i.e.* one request asks for a trip within the transportation time-frame of another one for the same trip. For instance, unitary transportation times imply *non-crossing requests*. With non-crossing requests, Max Flow WR can still be $2^M - M - 1$ times better than any feasible FCFS flow, where M is the number of stations.

For one vehicle and a given number of stations M, an instance reaching the $2^M - M - 1$ bound can be constructed as follows: The instance is based on a succession of repeated cyclic requests. A cyclic request is an ordered series of trip requests evolving along a cycle in the physical graph of stations. There are $2^M - M - 1$ cycles with different sets of stations and hence $2^M - M - 1$ different cyclic requests (we do not take the empty cycle nor cycles with only one station). Each cyclic request is repeated to have a total of q trip requests. The stations present in a cyclic request are called the support. Before each repetition of the same cyclic request, the entrance is forced into one specific station of the support, say s_1, thanks to a gadget that creates a request from every station to s_1. Then starts the first cyclic request that is special. It begins with s_1 and before each trip request of the cyclic request, there are a series of requests from its current origin station going out to every station not present in the support. The cyclic request is then repeated in order to contain in the end q trip requests. With one vehicle, on this instance, Max Flow can serve $(2^M - M - 1)q$ demands while any FCFS flow policy can serve at most $q + O(2^M)$. Asymptotically, when q tends to infinity, the gap between Max Flow WR and any FCFS flow tends to $2^M - M - 1$. For $M = 5$ stations, Figure 3.8 schemes how to create the requests for one repeated cyclic request which support is the set of 3 stations a, b and c.

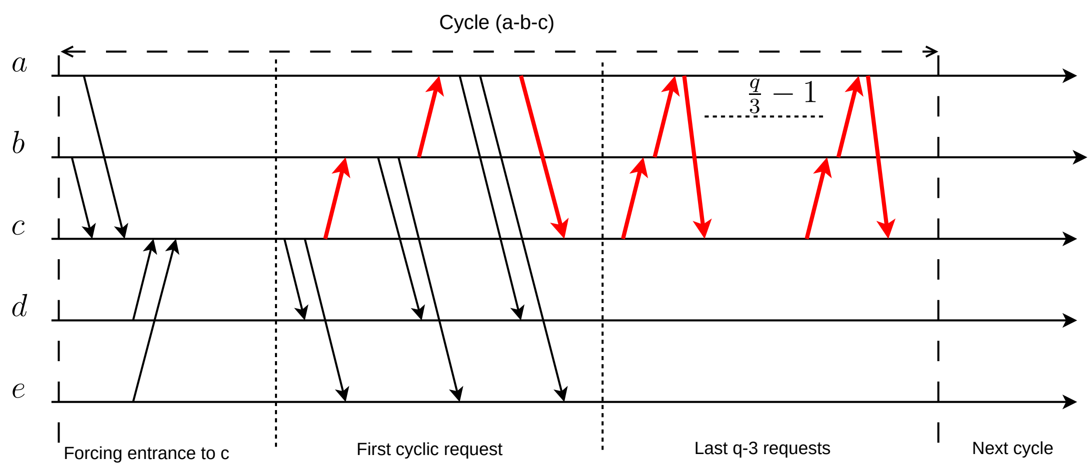


Figure 3.8: For non-crossing requests, the ratio between Max Flow With Reservation and any FCFS flow can be greater than $2^M - M - 1$.

3.5.2 An approximation algorithm for FCFS Flow 0/1 Trip Pricing

Previous sections schemed that Max Flow can be arbitrary far from a FCFS flow. We show here that with non crossing requests, and unitary maximum prices, the gap for pricing problems can be bounded. We present an approximation algorithm for FCFS Flow 0/1 Trip Pricing (FCFS Flow Trip Pricing with unitary maximum prices) for non crossing requests. To do so, first we give an approximation algorithm for FCFS Path 0/1 Trip Pricing which is the FCFS Flow 0/1 Trip Pricing problem with one vehicle. This approximation algorithm is based on the MAX FLOW optimal solution. It returns a *cyclic policy*, *i.e.* a policy that can serve only trip requests belonging to one oriented cycle in the spatial network.

Algorithm 1 FCFS PATH 0/1 TRIP PRICING Gready Approximation Algorithm

```
1: F* ← MAX FLOW solution for 1 vehicle in the time & space network;
2: for all Station s in path F* do                              ▷ Iterate on path F*
3:     if s is marked then          ▷ A cycle c (starting and ending at s) is detected
4:         n(c) ← n(c) + 1;
5:         Unmark all stations;
6:     end if
7:     Mark station s;
8: end for
9: return the cyclic policy defined by the cycle c with maximum value n(c)|c|.
```

Theorem 5. *Algorithm 1 provides a* $\frac{1}{(M+2)!}$*-approximation algorithm for the* FCFS PATH 0/1 TRIP PRICING *problem with non-crossing requests.*

Proof. Algorithm 1 gives, for each detected cycle c, its occurrence $n(c)$ and its length $|c|$ in the MAX FLOW optimum solution F^* for one vehicle. Figure 3.9 schemes an example of execution with 2 detected cycles each one appearing once. Each cycle has a length greater or equal to 2 and between two consecutive cycles we can iterate through at most $M-2$ stations (*lost requests*). It means that every M stations we detect at least a cycle of size 2. Hence, keeping only the detected cycles might lose a factor at most $2/M$:

$$\sum_c n(c)|c| \geq \frac{2}{M}|F^*|.$$

There are less than $M \times M!$ different cycles. Therefore the cycle c' with the maximum $n(c)|c|$ verifies:

$$n(c')|c'| \geq \frac{2}{M \times M \times M!}|F^*| \geq \frac{1}{(M+2)!}|F^*|.$$

Cycle c' defines a cyclic policy C' that provides at least the same gain ($C' \geq n(c')|c'|$) with a FCFS flow dynamic and all requests (assumed non-crossing). Finally, Algorithm 1 is polynomial, for non-crossing requests we have hence a $\frac{1}{(M+2)!}$-approximation on the optimal FCFS PATH 0/1 TRIP PRICING policy S^*:

$$C' \geq \frac{1}{(M+2)!}|F^*| \geq \frac{1}{(M+2)!}S^*.$$ □

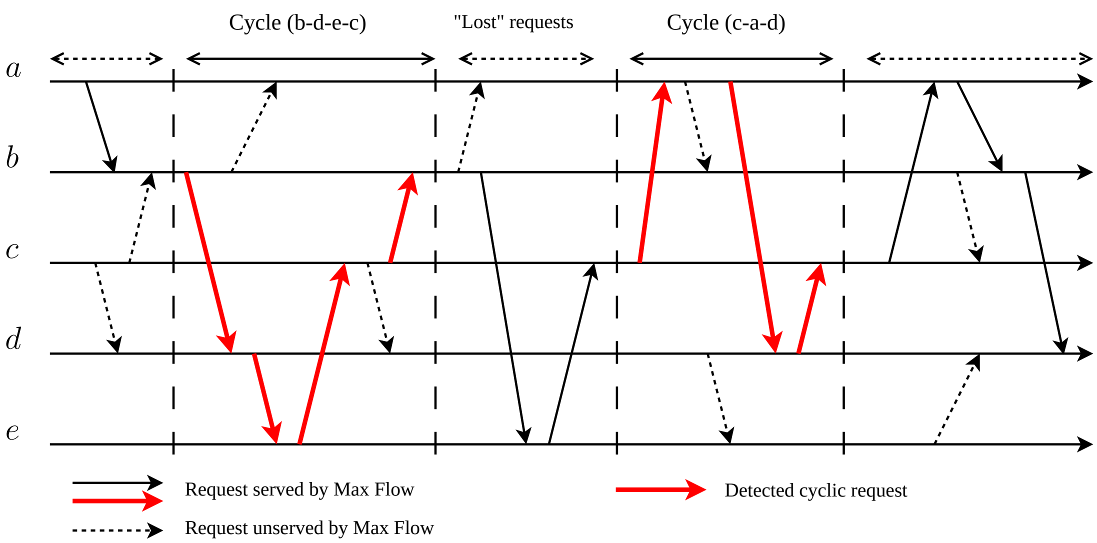


Figure 3.9: Example of execution of Greedy Algorithm 1 where two cycles are detected with occurrence 1.

We now extend the preceding FCFS path results to the FCFS flow problem.

Corollary 3. *For non-crossing requests we have the following results:*

- *Algorithm 1 provides a* $\frac{1}{N((M+2)!)}$*-approximation algorithm for the* FCFS FLOW 0/1 TRIP PRICING *problem.*
- *The approximability ratio of the* FCFS FLOW 0/1 TRIP PRICING *is within* $[\frac{1}{N((M+2)!)}, 39/40]$.
- *The worst case ratio between* MAX FLOW WITH RESERVATION *and any FCFS flow is within* $[2^M - M - 1, N((M+2)!)]$.

Proof. We assume non-crossing requests. Theorem 2 states that FCFS FLOW TRIP PRICING is not approximable within 39/40 even with unitary maximum prices, that is FCFS FLOW 0/1 TRIP PRICING.

Theorem 5 can be extended to any number of vehicles. Let $|F_1^*|$ be the MAX FLOW value for 1 vehicle and $|F_N^*|$ for N vehicles. Let S^* be the value of the optimal FCFS PATH 0/1 TRIP PRICING policy. We have $N|F_1^*| \geq |F_N^*| \geq S^*$ and hence $N((M+2)!)C' \geq S^*$. Therefore, Algorithm 1 provides a $\frac{1}{N((M+2)!)}$-approximation algorithm for the FCFS FLOW 0/1 TRIP PRICING problem and, unless P equals NP, FCFS FLOW 0/1 TRIP PRICING approximability ratio is within $[\frac{1}{N((M+2)!)}, 39/40]$.

Let $|F_N^{R*}|$ be the value of MAX FLOW WR for N vehicles. In the proof of Theorem 5, we saw that $C' \geq \frac{1}{(M+2)!}|F_1^*|$. Since $S^* \geq C'$ and $N|F_1^*| \geq |F_N^*| \geq |F_N^{R*}|$ we have: $N((M+2)!)S^* \geq |F_N^*|$. Moreover, we have seen in the previous section that there exists instances such that $|F_1^{R*}| \geq (2^M - M - 1)S^*$. Therefore the worst case ratio between MAX FLOW WITH RESERVATION and any FCFS flow is within $[2^M - M - 1, N((M+2)!)]$. □

3.6 Reservation in advance

For subscriptions to a periodic service, or for single requests asked far in advance, one can assume that users are ready to wait for an answer after expressing their requests. During this period, the system is able to consider several requests at the same time and to select which ones to serve in order to maximize the expected revenue or the number of trips sold.

Assuming no real-time hazards, this problem can be seen as deterministic. This request selection does not involve a FCFS flow constraint: it is a classic flow to optimize. Without considering user alternatives, we show in Section 3.6.1 that it amounts to solving a MAX FLOW problem, polynomially solvable. However, when considering spatial and temporal flexibilities, this request selection problem is equivalent to a MAX FLOW WITH ALTERNATIVE shown NP-hard in Section 3.6.

3.6.1 No flexibilities

When users have no flexibilities, they only want to take a specified trip. On a given horizon, considering a set of requests to take a trip between two specific stations at a specific time, we can represent all these requests on a time and space network. A MAX FLOW algorithm on this graph, with an amount of flow equal to the number of vehicles, solves the problem of which requests to accept. The MAX FLOW algorithm has a computational time polynomial in the number of stations and in the number of requests. Moreover, since all capacities on the arcs are integer the optimal solution will be "integral", *i.e.* a subset of trips to accept and not trip fractions.

3.6.2 Flexible requests

We consider now flexible requests where users are ready to change their origin and/or their destination stations, delay or advance the date of their trip. A user request can be satisfied by several station-to-station trip alternatives with possibly different gains. Each request can be arbitrarily accepted, *i.e.* served with one of its alternative, or refused. There is no consideration of a first come first served rule. The problem is to find the set of requests to serve in order to maximize the overall gain.

Max Flow With Alternative

- **Instance**: A set of stations S with capacities $\mathcal{K}_s$ for $s \in S$, a number N of vehicles with their distribution among the stations at the beginning of the horizon, a set $R = \{(s_o^{k,r}, t_o^{k,r}, s_d^{k,r}, t_d^{k,r}, p^{k,r}),\ k \in K,\ r \in R\}$ of trip requests with $|K|$ alternatives.
- **Solution**: The set of requests R' to serve with the alternative k chosen.
- **Measure**: The generated gain of the flow R':

$$\sum_{(r,k)\in R'} p^{k,r}.$$

Theorem 6. Max Flow With Alternative *is NP-hard even with requests of unitary price.*

Proof. We reduce the NP-hard problem 3-SAT to Max Flow With Alternative. We use a gadget called the "k-choices". It directs a flow of k vehicles from a station to exactly one station out of two. Figure 3.10 schemes an example for $k = 3$. The general construction is the following. There are k vehicles at station a to go either all to station b or c. At time step 0, there are k trip requests with no alternative to go from station a to stations $s_0 \dots s_{k-1}$ at time step 1. At time step 1, we can have a vehicle in each station $s_0 \dots s_{k-1}$. Then, there are k trip requests ($r_i,\ i \in \{0 \dots k-1\}$) with two alternatives: (1) to go from station $s_o^{1,i} = s_i$ to station $s_d^{1,i} = c$ or (2) to go from station $s_o^{2,i} = s_{i+1 \bmod k}$ to station $s_d^{2,i} = b$, arriving both at time step 2. The only possibility to serve all $2k$ trip requests is to accept either all trip alternatives (1) going to station b or all trip alternatives (2) going to station c. All other policies incur a loss of at least two trip requests.

We consider now a 3-SAT instance with m clauses and n literals. Each literal l is represented by 3 stations: $\dot{l}$ when the literal is unassigned, l when it is set to true and $\bar{l}$ when it is set to false. At the beginning of the horizon, there are m vehicles available at every station $\dot{l}$. At time step 0, there is a "k-choice" gadget with $k = m$ to direct a flow of m vehicles either to station l or $\bar{l}$. We create a station r to store the number of clauses satisfied (represented as the number of vehicles in station r at the end of the horizon). For each clause i (i=1 to m), there is a trip request at time step i with three alternatives. For clause $a \vee b \vee c$ the three alternative trips are to go from station a to station r, b to r or c to r.

The 3-SAT instance is satisfiable if and only if the Max Flow With Alternative instance serves $2mn + m$ demands: $2m$ for each of the n literal assignments (through a k-choice gadget) and m to satisfy all clauses. ∎

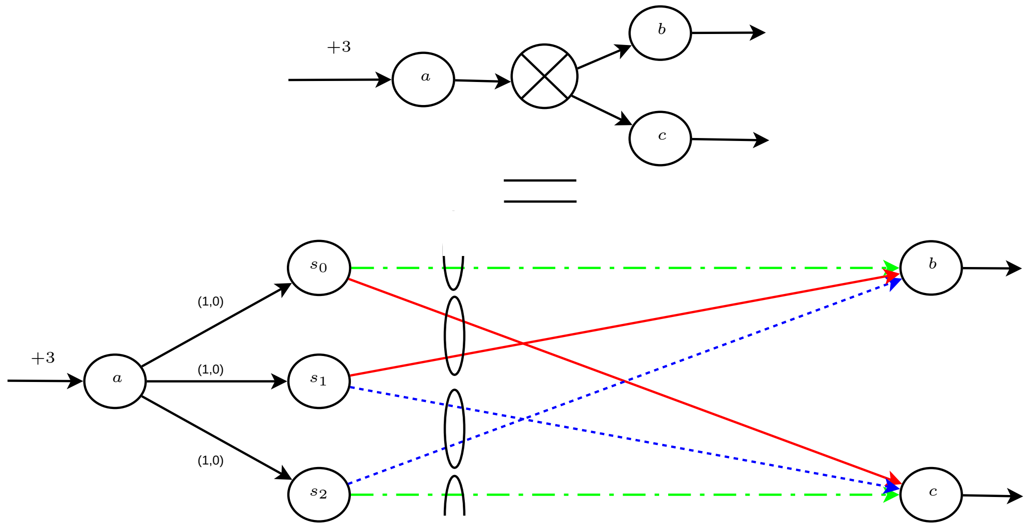


Figure 3.10: k-choices gadget, example with $k = 3$. On the upper part of the figure, the compact representation of the gadget.

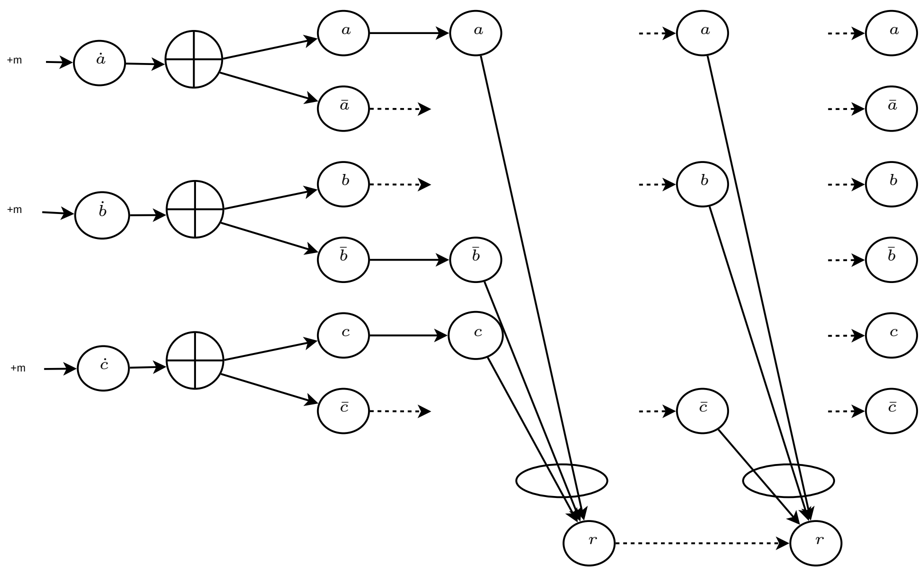


Figure 3.11: 3-SAT reduction as a MAX FLOW WITH ALTERNATIVE. Two clauses are represented: $a \vee \bar{b} \vee c$ and $a \vee b \vee \bar{c}$.

3.7 Conclusion

In this chapter, we have investigated a scenario-based approach for the VSS stochastic pricing problem. Its principle is to work a posteriori on a realization of the stochastic process: *a scenario*. Optimizing on a scenario provides heuristics and bounds for the stochastic problem. In this context, such approximation raises deterministic problems with a new constraint: the *First Come First Served constrained flow* (FCFS flow). We presented three such problems: 1) a system design problem, optimizing station capacity (FCFS Flow Station Capacities) and two operational problems setting static prices, 2) on the trips (FCFS Flow Trip Pricing), or 3) on the stations (FCFS Flow Station Pricing).

We showed that all three problems are APX-hard, *i.e.* inapproximable in polynomial time within a constant ratio. Therefore, we investigated a bound and an approximation algorithm using the Max Flow algorithm (hence relaxing the FCFS flow constraint). The theoretical guaranty (worst case) for the bound provided by the Max Flow algorithm on a scenario is exponential in the number of stations. Nevertheless, it is competitive in practice. We use Max Flow With Reservation to compute upper bounds in Chapter 6 devoted to the simulation. Moreover, from a theoretical point of view, it can be used to build a $\frac{1}{N((M+2)!)}$-approximation algorithm for the FCFS Flow Trip Pricing problem with unitary prices; with N the number of vehicles and M the number of stations.

We conjecture that the inapproximability ratios of FCFS Flow Trip/Station Pricing and FCFS Flow Station Capacities are greater than a factor linked to the number of stations. One can hence be satisfied to have an approximation algorithm that does not depend on the number of trip requests $|R|$. However, in current VSS, the number of trips sold in one day is in the order of M (or N). Therefore, an approximation algorithm in $|R|$ might be more useful.

Finally, giving good and usable heuristic solutions using scenario-based optimization, studying metaheuristic approaches might be interesting. However, it is not sure that they can explore such large space and provide good solutions within a reasonable time. Indeed, the evaluation cost of a *movement* on a static policy seems important, at first sight basically in the order of computing again the whole FCFS flow.

Chapter 4

Queuing Network Optimization with product forms

The art of doing mathematics consists in finding that special case which contains all the germs of generality.

David Hilbert (1862–1943)

Chapter abstract

This chapter proposes an approximation algorithm to solve a simpler stochastic VSS pricing problem than the general one presented in Chapter 2. In order to provide exact formulas and analytical insights: transportation times are assumed to be null, stations have infinite capacities and the demand is Markovian stationary over time. We propose a heuristic based on computing a MAXIMUM CIRCULATION on the demand graph together with a convex integer program solved optimally by a greedy algorithm. For M stations and N vehicles, the performance ratio of this heuristic is proved to be exactly $N/(N + M - 1)$. Hence, whenever the number of vehicles is large compared to the number of stations, the performance of this approximation is very good.

Keywords: Closed Queuing Networks; Pricing; Product forms; Continuous-time Markov decision process; Stochastic optimization; Approximation algorithms.

Résumé du chapitre

Ce chapitre propose un algorithme d'approximation pour résoudre un problème stochastique de tarification dans les systèmes de véhicules en libre service. Ce problème est simplifié par rapport à celui présenté Chapitre 2. De manière à obtenir des formules exactes et des résultats analytiques, les temps de transports sont considérés nulle, les stations ont des capacités infinis et la demande est markovienne stationnaire. Nous proposons une heuristique basée sur le calcul d'une CIRCULATION MAXIMUM sur le graphe des demandes couplé à un programme entier convexe résolu optimalement par un algorithme glouton. Pour M stations et N véhicules, le ratio de performance de cette heuristique est prouvé être exactement $N/(N+M-1)$. Par conséquent, lorsque le nombre de véhicules est grand devant le nombre de stations, la performance de cette approximation est très bonne.

Mots clés : Réseau de files d'attentes fermé ; Tarification ; Forme produit ; Processus de décision Markovien à temps continu ; Optimisation stochastique ; Approximation.

Contents

This chapter is based on the article "Pricing in Vehicle Sharing Systems: Queuing Network Optimization with product forms" (Waserhole and Jost, 2013a) submitted to the special issue on shared mobility systems in EURO Journal on transportation and logistics.

4.1 Introduction

In Chapter 2, Section 2.3.3, we discussed the properties of optimal dynamic and static policies. An optimal dynamic policy can be computed with an action decomposable Markov decision process. However the number of states of the MDP grows roughly as N^M, where N is the number of vehicles and M is the number of stations considered. This chapter proposes an approximation algorithm to solve a simpler stochastic VSS pricing problem than the general one presented in Chapter 2. In order to provide exact formulas and analytical insights: we investigate simplified stochastic models allowing an analytic formula for the performance evaluation of the system.

In Section 4.2, we define the simplified model we are going to restrain to. We consider VSS with stationary O-D demands and infinite station capacities, as in George and Xia (2011), but we also assume null transportation times. Under these assumptions, the VSS can be modeled as a closed queuing network of BCMP type. Its performance can therefore be computed analytically. We define static and dynamic stochastic pricing problems on such queuing networks.

In Section 4.3 we study a static heuristic policy provided by the Maximum Circulation on the demand graph. When the Maximum Circulation disconnects the city, vehicles have to be spread among the connected components. The vehicle distribution problem amounts to maximizing a separable concave function under linear and integrality constraints. It can be solved optimally by a greedy algorithm. The exact guaranty of performance of our heuristic on dynamic and static policies is proved to be $\frac{N}{N+M-1}$.

4.2 Simplified stochastic framework

We simplify the general stochastic framework defined in Chapter 2 in order to provide analytical results. In this chapter we restrict our study to a stationary demand and infinite station capacities as in George and Xia (2011) but also null transportation times. We focus on the objective of maximizing the number of trips sold by the system.

4.2.1 Simplified protocol

We consider a *real-time station-to-station protocol* as defined in Figure 4.1. A user asks for a vehicle at station a (here and now), with destination b. The system offers a price (or rejects the user = infinite price). The user either pays the price and the vehicle is transferred, or leaves the system.

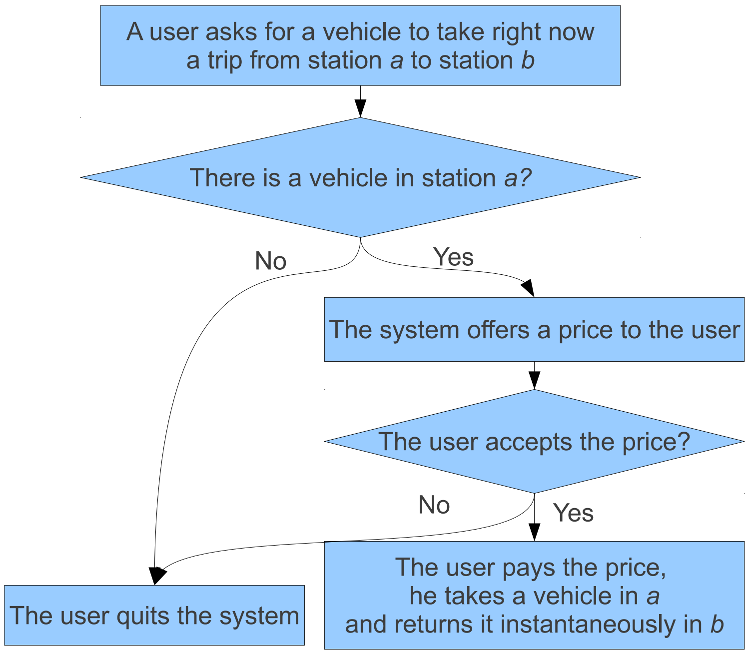


Figure 4.1: The real-time station-to-station protocol.

4.2.2 Simplified VSS stochastic evaluation model

Continuous-time Markov chain evaluation framework We model the VSS dynamic by a stochastic process: the *VSS stochastic evaluation model*. It measures VSS performances for a given policy (demand vector). We use this evaluation model to compare the performance of the proposed pricing policies in term of number of trips sold. We now define formally the VSS stochastic evaluation model under the real-time station-to-station protocol (defined in Figure 4.1).

VSS Stochastic Evaluation Model

- **Input:**
  - A number N of vehicles and a set $\mathcal{M}$ of stations:
    - A set $\mathcal{S}$ of states: $\mathcal{S} = \left\{ \left(n_a : \ a \in \mathcal{M} \right) \ / \ \sum_{a \in \mathcal{M}} n_a = N \right\}$;
    - State $s = (n_a : \ a \in \mathcal{M})$ represents the vehicle distribution in the city space: n_a is the number of vehicles in station $a \in \mathcal{M}$.
  - A policy λ:
    - $\lambda^s_{a,b}$ is the arrival rate of users to take the trip $(a, b) \in \mathcal{D} = \mathcal{M} \times \mathcal{M}$, between state $s = (\ldots, \ n_a \geq 1, \ldots, \ n_b, \ldots) \in \mathcal{S}$ and state $(\ldots, \ n_a - 1, \ldots, \ n_b + 1, \ldots) \in \mathcal{S}$;
    - The graph spanned by $\left\{ s \in \mathcal{S}, \ (a, b) \in \mathcal{D}, \ \lambda^s_{a,b} > 0 \right\}$ is supposed to be strongly connected.
- **Output:** The expected number of trips sold in the steady state behavior of the continuous-time Markov chain defined by states $\mathcal{S}$ and transition rates λ.

Notice that the number of states is exponential in the number of vehicles and stations (see Proposition 1 page 43). For instance, for a system with $N = 150$ vehicles and $M = 50$ stations there are already $\binom{N+M-1}{N} \simeq 10^{47}$ states!

Steady-state distribution of the continuous-time Markov chain For any strongly connected dynamic policy, the unique stationary distribution π over the state space $\mathcal{S}$ of the continuous-time Markov chain with transition rate λ satisfies Equations (4.1) (Puterman, 1994). Let e_a be the unit vector for component $a \in \mathcal{M}$: $e_a = (0, \ldots, 0, n_a = 1, 0, \ldots, 0)$.

$$\sum_{s \in \mathcal{S}} \pi_s = 1, \tag{4.1a}$$

$$\sum_{\substack{(a,b) \in \mathcal{D} \\ s - e_a + e_b \in \mathcal{S}}} \pi_s \lambda^s_{a,b} = \sum_{\substack{(b,a) \in \mathcal{D}, \ s' \in \mathcal{S} \\ s' - e_b + e_a = s}} \pi_{s'} \lambda^{s'}_{b,a}, \qquad \forall s \in \mathcal{S}, \tag{4.1b}$$

$$\pi_s \geq 0, \qquad \forall s \in \mathcal{S}. \tag{4.1c}$$

Closed queuing network model for static policies The VSS stochastic evaluation model can be represented as a closed queueing network for static policies. An example with 2 stations is schemed in Figure 4.2. This closed queuing network is built as follows.

Since there is a fixed number of vehicles circulating in the network, it is natural to see the system from a vehicle's perspective. Each station $a \in \mathcal{M}$ is represented by a server a with infinite capacity queue. The N vehicles are N jobs waiting in these queues for users to take them. The service rate λ_a of server a is equal to the average number of users willing to take a vehicle at station a: $\lambda_a = \sum_{(a,b)\in\mathcal{D}} \lambda_{a,b}$. A vehicle taken by a user for a trip $(a, b) \in \mathcal{D}$ is represented by a job processed by server a with routing probability $\frac{\lambda_{a,b}}{\lambda_a}$. When a vehicle (a job) is taken for the trip (a, b) it is transferred instantaneously in station (buffer) b.

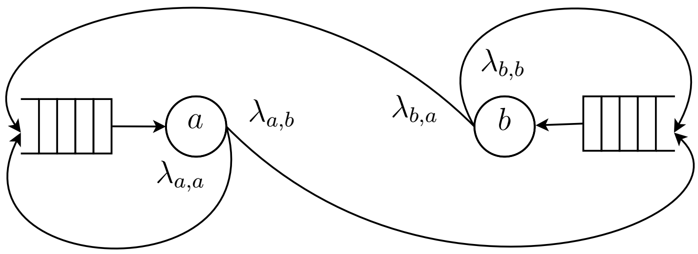


Figure 4.2: A closed queuing network model: servers represent users demands.

Analytic evaluation for static policies The stochastic evaluation model for static policies is the same as the one considered by George and Xia (2011) but with null transportation times. They provide a compact form to compute the system performance using the BCMP network theory (Baskett *et al.*, 1975). In Section 4.3.2, we consider static policies providing demands for which the performance evaluation is slightly simpler than the formula of George and Xia (2011), see Lemma 1.

An important concept that we use for a static policy (with demand λ) is the *availability* A_a of (a vehicle at) station $a \in \mathcal{M}$ which is the probability that station a contains at least one vehicle. Availibilities satisfy steady-state equations:

$$\sum_{b\in\mathcal{M}} A_a \lambda_{a,b} = \sum_{b\in\mathcal{M}} A_b \lambda_{b,a}, \qquad \forall a \in \mathcal{M}. \tag{4.2}$$

Notice that availibilities are not totally determined by (4.2) because they also depend on the number of vehicles.

4.2.3 Simplified VSS stochastic pricing problem

We now define formally the problem we tackle in this chapter.

VSS Stochastic Continuous Pricing Transit Maximization

- **Instance**:
  - A number N of vehicles available;
  - A set $\mathcal{M}$ of stations with infinite capacities;
  - The maximum demand per time unit $\Lambda_{a,b}$ to take every trip $(a, b) \in \mathcal{D}$.
- **Solution**:

  [**Dynamic Policy**] A demand $\lambda_{a,b}(s) \in [0, \Lambda_{a,b}]$, to take each trip $(a, b) \in \mathcal{D}$ function of the system's state $s \in \mathcal{S}$.

  [**Static Policy**] A tuple $(\lambda,\ k,\ \vec{\mathcal{M}},\ \vec{N})$, where:
  - $\lambda_{a,b} \in [0, \Lambda_{a,b}]$ is the demand to take each trip $(a, b) \in \mathcal{D}$,
  - λ defines a set of strongly connected components $\vec{\mathcal{M}} = \{\mathcal{M}_1, \ldots, \mathcal{M}_k\}$,
  - $\vec{N} = (N_1, \ldots, N_k)$ is the vehicle distribution over $\vec{\mathcal{M}}$, $(\sum_{i=1}^{k} N_i = N)$.
- **Measure**: The expected number of trips sold of the pricing policy measured by the stochastic evaluation model.

We restrict the study of dynamic policies to the (dominant) class for which the graph spanned by $\left\{(a, b \in \mathcal{M}, s \in \mathcal{S},\ \lambda^s_{a,b} > 0\right\}$ has only one strongly connected component. Otherwise, the stationary distribution on the state graph is not unique: it depends on the initial state of the system.

Sometimes optimal static policies need more than one strongly connected components on the station graph. An example is given in Proposition 5 page 56. The k strongly connected components of the static policy graph $G(\mathcal{M}, \lambda)$ divides the city into k independent VSS, sharing a number N of vehicles. The vehicle distribution has then to be explicitly specified since it impacts the policy performance. For dynamic policies, the vehicle distribution is explicit (defined by the system states for single component policies). That is why for ease of notations the stochastic evaluation model is defined for dynamic policies (any static policy can be represented as a dynamic one).

4.2.3.1 Complexity in this simplified stochastic framework

The discussion on complexity of Section 2.3.2, page 50, for the general VSS stochastic pricing problem can be adapted to this simplified problem.

To tackle large scale (real-world) systems, we need solution methods that have computational time polynomial in N and M. The solutions (pricing policies) produced (output) need also to be of moderate size. Notice that the state graph (of exponential size) representing all possible vehicle distributions (system's states) is not part of the problem input. The explicit representation of dynamic policies is hence not tractable.

For static policies, measuring exactly the stochastic evaluation model is polynomial in M and N: George and Xia (2011) provide a product form formula and algorithms to compute the stochastic evaluation model for a static pricing policy. However, we are able to prove that the decision version of the above static pricing problem is in NP only under further assumptions (see Section 2.3.2, page 50).

We discussed in Section 2.3.3, page 51, the problem of characterizing dynamic and static optimal policies. The complexity is unknown for both classes of policies. The deterministic version of the stochastic pricing problem was shown NP-hard in Chapter 3. Nevertheless there is no obvious reduction between these problems[1].

4.3 Maximum Circulation approximation

In this section we study an approximation algorithm based on the Maximum Circulation problem (Edmonds and Karp, 1972): a network flow problem with flow conservation at all nodes (no source no sink).

4.3.1 Maximum Circulation Upper Bound

A vector λ is called a *circulation* if it is solution of the following LP.

Maximum Circulation LP

$$\begin{aligned} \max \quad & \sum_{(a,b)\in\mathcal{D}} \lambda_{a,b} & \\ \text{s.t.} \quad & \sum_{(a,b)\in\mathcal{D}} \lambda_{a,b} = \sum_{(b,a)\in\mathcal{D}} \lambda_{b,a}, & \forall a \in \mathcal{M}, \\ & 0 \leq \lambda_{a,b} \leq \Lambda_{a,b}, & \forall (a,b) \in \mathcal{D}. \end{aligned}$$

Theorem 7. *The objective value of* Maximum Circulation *on the demand graph is an upper bound on any dynamic policy for any number of vehicles.*

Proof. From any dynamic policy, with transition rate $\lambda^s_{a,b} \leq \Lambda_{a,b}$ in state $s \in \mathcal{S}$ for trip $(a,b) \in \mathcal{D}$, we construct a circulation on the demand graph with same value. Under this policy, the stationary distribution π over the state space $\mathcal{S}$ of the continuous-time Markov chain defined by λ satisfies Equations (4.1). Let $\lambda'_{a,b}$ be the

1. The stochastic version restricts to exponential distributions, and not general time-dependent distributions.

expected transit for any trip $(a,b) \in \mathcal{D}$: $\lambda'_{a,b} = \sum_{s\in\mathcal{S}} \pi_s \lambda^s_{a,b}$. We show that λ' is a circulation. The capacity constraints are satisfied since $\sum_{s\in\mathcal{S}} \pi_s = 1$ and hence:

$$\lambda'_{a,b} = \sum_{s\in\mathcal{S}} \pi_s \lambda^s_{a,b} \leq \sum_{s\in\mathcal{S}} \pi_s \Lambda_{a,b} = \Lambda_{a,b}, \quad \forall (a,b) \in \mathcal{D}.$$

Flow conservation constraints are satisfied because in the steady state of a dynamic policy, a station receives as many vehicles as it is sending. Finally, the expected transit of the system is equal to $\sum_{(a,b)\in\mathcal{D}} \lambda'_{a,b}$ which is the value of circulation λ'. $\square$

4.3.2 Maximum Circulation static policy

The Maximum Circulation outputs a demand vector $\lambda \leq \Lambda$. It is natural to try to use this demand vector as a static policy. However, whenever the Maximum Circulation is not strongly connected, one has to specify a vehicle distribution $\vec{N}$ over the k strongly connected component $\vec{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_k\}$. In Proposition 6 we show that this issue may indeed occur. We call a static policy $\phi = (\lambda,\ k,\ \vec{\mathcal{M}},\ \vec{N})$ a *circulation policy* if λ is a circulation.

Proposition 6. *The optimal solution(s) of* Maximum Circulation *might consist of more than one strongly connected component.*

Proof. Consider the demand graph in Figure 4.3 consisting of $\Lambda = 1$ for all drawn arcs (both dotted and straight). The unique Maximum Circulation sets $\lambda = 1$ for straight arcs and 0 elsewhere. Its policy demand graph is not strongly connected. $\square$

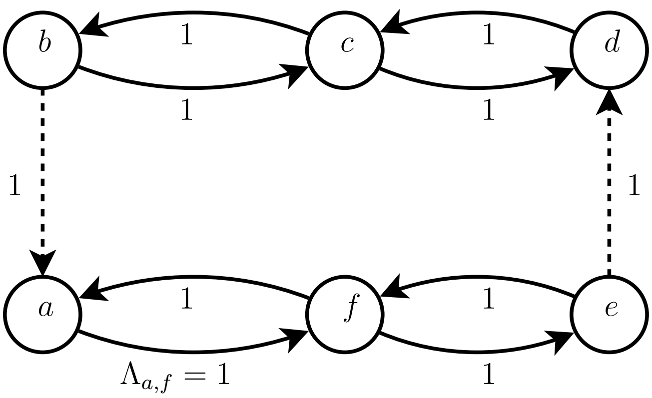


Figure 4.3: Maximum Circulation can consist of several strongly connected components.

4.3.2.1 Evaluation for a given vehicle distribution

Recall that for a static policy ϕ, the availability $A_a(\phi)$ of (a vehicle at) station $a \in \mathcal{M}$ is the probability that station a contains at least one vehicle. Moreover, to any static policy $\phi = (\lambda,\ k,\ \vec{\mathcal{M}},\ \vec{N})$ is associated a Continuous-Time Markov Chain, CTMC(ϕ), that is used for its evaluation.

Lemma 1 explains how to compute the expected transit of a circulation policy. It essentially says that the availability of a station is $\frac{N}{N+M-1}$ for a circulation spanning only one strongly connected component with M stations.

Lemma 1. *For any circulation λ and any vehicle distribution $\vec{N}$, the expected transit $T(\phi)$ of the circulation policy $\phi = (\lambda,\ k,\ \vec{\mathcal{M}},\ \vec{N})$ is equal to:*

$$T(\phi) = \sum_{i=1}^{k} \left(\frac{N_i}{N_i + |\mathcal{M}_i| - 1} \sum_{a,b \in \mathcal{M}_i} \lambda_{a,b} \right).$$

The remaining of Section 4.3.2.1 is devoted to a proof of Lemma 1. It is done by expressing relations between transit, availability and the continuous-time Markov chain formulation.

Lemma 2. *For a static policy ϕ with a given vehicle distribution, the stationary distribution π over the states of the continuous-time Markov chain CMTC(ϕ) is unique.*

Proof. A Markov chain is said to be *irreducible* if its state space is a single communicating class (a single strongly connected component); in other words, if it is possible to get to any state from any state. The continuous-time Markov chain CMTC(ϕ) defined by a static policy ϕ is irreducible, therefore there is a unique stationary distribution (Puterman, 1994). □

The availability $A_a(\pi)$ of station $a \in \mathcal{M}$ is equal to the sum of the stationary distributions π_s of the states $s \in \mathcal{S}$ where there is at least one vehicle in station a:

$$A_a(\pi) := \sum_{s=(\dots,n_a \geq 1,\dots) \in \mathcal{S}} \pi_s. \tag{4.3}$$

Since for any static policy ϕ, a stationary distribution π can be computed on CTMC(ϕ), for convenience we also denote:

$$A_a(\phi) := A_a\big(\pi(\phi)\big).$$

The expected transit $T(\phi)$ of static policy ϕ is then:

$$T(\phi) = \sum_{a \in \mathcal{M}} \left(A_a(\phi) \sum_{b \in \mathcal{M}} \lambda_{a,b} \right).$$

We now state a couple of lemmas that combined will prove Lemma 1.

Lemma 3. *For a static policy ϕ, CTMC(ϕ) is the product of k independent CTMC(ϕ^i), where $\phi^i = (\lambda_{(a,b)\in\mathcal{M}_i^2},\ 1,\ \{\mathcal{M}_i\},\ (N_i))$ is a static policy with one single strongly connected component. The expected transit $T(\phi)$ is then decomposed as follows:*

$$T(\phi) = \sum_{a\in\mathcal{M}} \left(A_a(\phi) \sum_{b\in\mathcal{M}} \lambda_{a,b} \right) = \sum_{i=1}^{k} \sum_{a\in\mathcal{M}_i} \left(A_a(\phi^i) \sum_{b\in\mathcal{M}_i} \lambda_{a,b} \right).$$

□

An invariant measure of a CTMC is a stationary distribution associated with some initial distribution (over the states of the chain). From Lemma 2, static policies have a unique stationary distribution. For strongly connected circulation policies there exists only a unique invariant measure. However, for disconnected circulation policies there exist several invariant measures.

The following lemma will be used both to prove Lemma 1 but also for the purpose of Section 4.3.3.2. We denote by $\mathcal{S}(N,M)$ the state set of all distributions of N vehicles among M stations.

Lemma 4. *For any circulation λ, $\pi_s = \frac{1}{|\mathcal{S}(N,M)|}$, $\forall s \in \mathcal{S}(N,M)$, is an invariant measure of the stationary distribution of the continuous-time Markov chain defined by states $\mathcal{S}(N,M)$ and transition rates λ.*

Proof. Let $\lambda_a^+ = \sum_{b\in\mathcal{M}} \lambda_{a,b}$ and $\lambda_a^- = \sum_{b\in\mathcal{M}} \lambda_{b,a}$. Since λ is a circulation we have $\lambda_a^+ = \lambda_a^-$. Let $\delta^+(s)$ (resp. $\delta^-(s)$) be the sum of the outgoing (resp. incoming) transition rates on state $s = (n_a :\ a \in \mathcal{M}) \in \mathcal{S}(N,M)$, we have:

$$\delta^+(s) = \sum_{\substack{(a,b)\in\mathcal{D} \\ s-e_a+e_b\in\mathcal{S}(N,M)}} \lambda_{a,b}^s = \sum_{a\in\mathcal{M}\ |\ n_a>0} \lambda_a^+,$$

and

$$\delta^-(s) = \sum_{\substack{(b,a)\in\mathcal{D},\ s'\in\mathcal{S}(N,M) \\ s'-e_b+e_a=s}} \lambda_{b,a}^{s'} = \sum_{a\in\mathcal{M}\ |\ n_a>0} \lambda_a^-.$$

Therefore $\delta^+(s) = \delta^-(s)$ and hence $\pi_s = \frac{1}{|\mathcal{S}(N,M)|}$, $\forall s \in \mathcal{S}(N,M)$, is solution of the stationary distribution Equations (4.1) of the continuous-time Markov chain with states $\mathcal{S}(N,M)$ and transition rates λ:

$$\sum_{\substack{(a,b)\in\mathcal{D} \\ s-e_a+e_b\in\mathcal{S}(N,M)}} \pi_s \lambda_{a,b}^s = \sum_{\substack{(b,a)\in\mathcal{D},\ s'\in\mathcal{S}(N,M) \\ s'-e_b+e_a=s}} \pi_{s'} \lambda_{b,a}^{s'}, \qquad \forall s \in \mathcal{S}(N,M),$$

$$\sum_{s\in\mathcal{S}(N,M)} \pi_s = 1,$$

$$\pi_s \geq 0, \qquad \forall s \in \mathcal{S}(N,M).$$

□

Lemma 5. *For the uniform stationary distribution $\pi_s = \frac{1}{|\mathcal{S}(N,M)|}$, $s \in \mathcal{S}(N, M)$, the availability of any station is equal to $\frac{N}{N+M-1}$.*

Proof. From Proposition 1, the number of distributions of N vehicles among M stations is equal to $|\mathcal{S}(N, M)| = \binom{N+M-1}{N}$. For any station $a \in \mathcal{M}$, there are $|\mathcal{S}(N-1, M)|$ states with at least one vehicle available in station a. If each state has the same stationary distribution, $\pi_s = \frac{1}{|\mathcal{S}(N,M)|}$, $s \in \mathcal{S}(N, M)$, computing the availability $A(\pi)$ of a vehicle at any station (Equation (4.3)) amounts to computing a ratio between two numbers of states:

$$A(\phi) = \frac{|\mathcal{S}(N-1, M)|}{|\mathcal{S}(N, M)|} = \frac{\binom{N+M-2}{N-1}}{\binom{N+M-1}{N}} = \frac{\frac{(N+M-2)!}{(N-1)!(M-1)!}}{\frac{(N+M-1)!}{(N)!(M-1)!}} = \frac{N}{N+M-1}.$$ □

Lemma 6. *For a circulation policy ϕ and for any strongly connected component $\mathcal{M}_i$, the availability $A(\phi^i)$ of a vehicle at any station $a \in \mathcal{M}_i$ is equal to:*

$$A(\phi^i) = \frac{N_i}{N_i + |\mathcal{M}_i| - 1}.$$

Proof. Combining Lemma 2 and 4, the unique stationary distribution over the states $\mathcal{S}(N_i, M_i)$ of CTMC(ϕ^i) for any circulation policy $\phi^i = (\lambda_{(a,b)\in\mathcal{M}_i^2},\ 1,\ \{\mathcal{M}_i\},\ (N_i))$ is $\pi_s = \frac{1}{|\mathcal{S}(N_i,M_i)|}$, $s \in \mathcal{S}(N_i, M_i)$. We can hence apply Lemma 5 to conclude.

□

Proof of Lemma 1. Combine Lemma 3 and 6. □

4.3.2.2 Optimality of the greedy distribution of vehicles

Let $\{\mathcal{M}_1, \ldots, \mathcal{M}_k\}$ be the set of the k strongly connected components of a circulation λ. If we allocate N_i vehicles to component i, the expected transit of the policy $\phi^i = (\lambda_{(a,b)\in\mathcal{M}_i^2}, 1, \{\mathcal{M}_i\}, \{N_i\})$ is:

$$T(\phi^i) = f_i(N_i) = \frac{N_i}{N_i + M_i - 1} \sum_{a,b\in\mathcal{M}_i} \lambda_{a,b}. \tag{4.4}$$

For a distribution $\vec{N} = (N_1, \ldots, N_k)$ of the N vehicles, the expected transit of policy $\phi = (\lambda, k, \vec{\mathcal{M}}, \vec{N})$ is hence:

$$T(\phi) = f(\vec{N}) = \sum_{i=1}^{k} f_i(N_i). \tag{4.5}$$

The optimal distribution $\vec{N}^*$ of the N vehicles among the k strongly connected components is then solution of the following problem:

$$\begin{aligned} \vec{N}^* = \max\ & f(\vec{N}) \\ \text{s.t.}\ & \sum_{i=1}^{k} N_i = N, \\ & \vec{N} \in \mathbb{Z}_+^k. \end{aligned}$$

Consider the following algorithm for finding a feasible solution to the previous problem:

Algorithm 2 Greedy algorithm for load distribution

1: $\vec{N} := (0, \dots, 0)$
2: **for** $n = 1$ to N **do**
3: Choose $j \in \arg\max_{i \in \{1,\dots,k\}} f(\vec{N} + e_i)$;
4: $\vec{N} := \vec{N} + e_j$;
5: **end for**
6: **return** $\vec{N}$.

In general Algorithm 2 may not provide an optimal solution. A function $f(\vec{N})$ for which there exist functions f_i such that $\forall \vec{N},\ f(\vec{N}) = \sum_{i=1}^{k} f_i(N_i)$, is called separable. Moreover if each f_i is concave, f is called concave separable.

Separable concave functions are of interest in mathematical economics, an example is the gain function (4.5). It turns out that separable concavity is enough for the greedy algorithm to find an optimal solution under the constraint $\sum_{i=1}^{k} N_i = N$ (see Theorem 8). Maximizing separable concave functions can also be done over more complex feasible spaces, such as polymatroids (Glebov, 1973; Shenmaier, 2003).

Theorem 8. *Let k be a positive integer, $\{f_i\}_{i \in \{1,\dots,k\}}$ be concave functions and $N \in \mathbb{Z}_+$. Also denote $f(\vec{N}) := \sum_i f_i(N_i)$. Then the solution of the following integer program is attained by greedy Algorithm 2.*

$$\begin{aligned} \max & \sum_{i=1}^{k} f_i(N_i) \\ s.t. & \sum_{i=1}^{k} N_i = N, \\ & \vec{N} \in \mathbb{Z}_+^k. \end{aligned}$$

Proof. We give a proof by induction on N. The case $N = 0$ is trivial since $\vec{N} = (0, \ldots, 0)$ is the only feasible solution. Assume case N is correct: the greedy algorithm provides an optimal solution, say $\vec{N}^*$ for N. Now, let $\vec{N}'$ be an optimal solution for $N + 1$. Choose $j \in \{1, \ldots, k\}$ such that $N'_j > N^*_j$. By induction hypothesis, $f(\vec{N}^*) \geq f(\vec{N}' - e_j)$. Also, by concavity of f_j and because $N'_j - 1 \geq N^*_j$, one has:

$$\begin{aligned} f(\vec{N}^* + e_j) &= f(\vec{N}^*) + f_j(N^*_j + 1) - f_j(N^*_j) \\ &\geq f(\vec{N}^*) + f_j(N'_j) - f_j(N'_j - 1) \\ &\geq f(\vec{N}' - e_j) + f_j(N'_j) - f_j(N'_j - 1) = f(\vec{N}'). \end{aligned}$$

A solution found by the greedy algorithm is hence at least as good as $f(\vec{N}^* + e_j)$ which is at least as good as $f(\vec{N}')$. □

Corollary 4. *For any fixed λ and any $N \in \mathbb{Z}_+$, a vehicle distribution $\vec{N} \in \mathbb{Z}_+^{k(\lambda)}$ maximizing the expected transit under the constraint $\sum_{i=1}^{k} N_i = N$ can be computed with greedy Algorithm 2.*

Proof. Let $\{\mathcal{M}_1, \ldots, \mathcal{M}_k\}$ be the set of the strongly connected components of the static policy graph $G(\mathcal{M}, \lambda)$. For any static policy, the expected transit of the system is the sum of the expected transit of each component, hence the gain function is separable. The concavity of the gain function in each component can be deduced from (4.4) for circulation policies, and is proved in (George and Xia, 2011, Theorem 2) for general static policies. □

4.3.3 Performance evaluation

We study the performance of the Maximum Circulation static policy together with its optimal vehicle distribution.

4.3.3.1 An upper bound on the approximation ratio

The expected transit of the Maximum Circulation static policy together with its optimal vehicle distribution can be arbitrarily close to $\frac{N}{N+M-1}$ times the value of a static policy:

Proposition 7. *For any number $M \geq 2$ of stations and any number N of vehicles, the ratio between the value of* Maximum Circulation *policy and a static policy can be arbitrary close to $\frac{N}{N+M-1}$.*

Proof. We consider instances with N vehicles, $M \geq 2$ stations $\mathcal{M} = \{1, \ldots, M\}$ and demand graph consisting of a circuit $\{1, \ldots, M, 1\}$ with maximum demand $\Lambda_{i,i+1} = k,\ i \in \{1, \ldots, M-1\}$ and $\Lambda_{M,1} = 1$ (all other demands are equal to 0).

The Maximum Circulation policy opens all trips of the circuit to 1. Its value P_{Circ^*} is equal to: $P_{Circ^*} = \frac{NM}{N+M-1}$.

Consider the generous static policy opening all trips to their maximum value: $\lambda = \Lambda$. The generous static policy demand graph is a circuit, hence the expected transit $(A_a \times \Lambda_{a,b})$ is the same for all trips (a, b) of the circuit. Availabilities A satisfy Equations (4.2) hence:

$$A_M \times 1 = A_i \times k, \quad \forall i \in \{1, \ldots, M-1\}, \text{ so:}$$

$$\sum_{a \in \mathcal{M}} A_a = A_M \left(1 + \frac{M-1}{k}\right).$$

Since $\sum_{a \in \mathcal{M}} A_a = 1$ for one vehicle, and $\forall a \in \mathcal{M},\ A_a$ is a non decreasing function of the number of vehicles (George and Xia, 2011), we have that $\sum_{a \in \mathcal{M}} A_a \geq 1$. Hence, $\lim_{k \to \infty} A_M(k) = 1$ and $\lim_{k \to \infty} A_i(k) = 0, \quad \forall i \in \{1, \ldots, M-1\}$. When $k \to \infty$, the value of the generous static policy is then $\lim_{k \to \infty} P_{Gen}(k) = M$.

The ratio between the static generous policy and the Maximum Circulation static policy can then be arbitrary close to:

$$\frac{N}{N+M-1} = \lim_{k \to \infty} \frac{P_{Gen}(k)}{P_{Circ^*}(k)}. \qquad \square$$

4.3.3.2 A tight guaranty of performance

Actually, the $\frac{N}{N+M-1}$ upper bound of Proposition 7 is the exact ratio of performance of Maximum Circulation static policy together with its optimal vehicle distribution:

Theorem 9. Maximum Circulation *static policy together with its optimal vehicle distribution is a tight $\frac{N}{N+M-1}$-approximation on both static and dynamic optimal policies.*

To the best of our knowledge, it is not easy to prove that Maximum Circulation static policy together with the optimal deterministic vehicle distribution is a $\frac{N}{N+M-1}$-approximation. Therefore we use a probabilistic proof (Lemma 8) that essentially says that the expected availability of a circulation policy with a specific random vehicle distribution is at least $\frac{N}{N+M-1}$, which means that a circulation policy with its optimal vehicle distribution has at least this performance. Still, before proving this results, we need to state another lemma on random vehicle distribution policies.

For a random distribution of vehicles $\vec{N}^R$, and a static policy λ with k strongly connected components $\vec{M}$, let $\phi^R = (\lambda, k, \vec{M}, \vec{N}^R)$ be the associated random vehicle distribution static policy and let $\pi^R(\phi^R)$ be the stationary distribution over the states of CMTC(ϕ^R).

Lemma 7. *The stationary distribution $\pi^R(\phi^R)$ over the CMTC(ϕ^R) defined by a static policy ϕ^R with random vehicle distribution $\vec{N}^R$ is unique.*

Proof. Recall that $\pi(\phi)$ is the stationary distribution over the states of the CMTC(ϕ) associated to static policy ϕ with deterministic vehicle distribution. We have:

$$\pi_s^R(\phi^R) := \sum_{(N_1,\dots,N_k)\ /\ \sum_{j=1}^k N_j = N} \mathbb{P}\Big(\vec{N} = (N_1,\dots,N_k)\Big) \times \pi_s\Big(\lambda, k, \vec{M}, (N_1,\dots,N_k)\Big).$$

From Lemma 2, for any deterministic vehicle distribution static policy ϕ, $\pi(\phi)$ is unique. Therefore the stationary distribution is also unique for any random vehicle distribution static policy. $\square$

Consider the random distribution $\vec{N}^U$ of vehicles to components induced by the uniform distribution on $\mathcal{S}(N, M)$ of vehicles among stations: For any vehicle distribution $\vec{N} = (N_1,\dots,N_k)$, the probability that $\vec{N}^U$ allocates $(N_1,\dots,N_k)$ equals:

$$\mathbb{P}\Big(\vec{N}^U = (N_1,\dots,N_k)\Big) := \frac{\Big|\Big\{(n_a:\ a\in\mathcal{M}) \in \mathcal{S}(N,M)\ /\ \forall i\in\{1,\dots,k\},\ \sum_{a\in M_i} n_a = N_i\Big\}\Big|}{|\mathcal{S}(N,M)|}.$$

Let ϕ^U be the random static circulation policy defined by the random uniform distribution $\vec{N}^U$ over $\mathcal{S}(N, M)$.

Lemma 8. *Let $N, M > 0$ and λ be a circulation with k strongly connected components ($\sum_{i=1}^k |\mathcal{M}_i| = M$). For any random uniform vehicle distribution circulation policy $\phi^U = (\lambda, k, \vec{\mathcal{M}}, \vec{N}^U)$, the availability $A_a(\phi^U)$ of a vehicle at any station $a \in \mathcal{M}$ is $\frac{N}{N+M-1}$. In other words, $\forall i \in \{1,\dots,k\}$, $\forall a \in \mathcal{M}_i$:*

$$A_a(\phi^U) := \sum_{(N_1,\dots,N_k)\ /\ \sum_{j=1}^k N_j = N} \mathbb{P}\Big(\vec{N}^U = (N_1,\dots,N_k)\Big) \times \frac{N_i}{N_i + M_i - 1} = \frac{N}{N+M-1}.$$

Proof. From Lemma 4, for any random uniform vehicle distribution circulation policy ϕ^U, $\pi_s(\phi^U) = \frac{1}{\mathcal{S}}$, $\forall s \in \mathcal{S}(N,M)$, is an invariant measure of CMTC(ϕ^U).

Moreover from Lemma 7, for any random vehicle distribution circulation policy, there exits a unique stationary distribution over the states of the CMTC(ϕ^U). Therefore for any random uniform vehicle distribution policy, the stationary distribution is $\pi_s(\phi^U) = \frac{1}{\mathcal{S}}$, $\forall s \in \mathcal{S}(N,M)$.

Finally we can apply Lemma 5 to conclude that $A_a(\phi^U) = \frac{N}{N+M-1}$. $\square$

Remark 5. *The previous proof is somewhat magical: It avoids computing the average over all vehicle distributions of the availability that does not seem to collapse to closed form formula.*

We can now prove the approximation ratio of MAXIMUM CIRCULATION static policy together with its optimal vehicle distribution.

proof of Theorem 9. Let $Circ^*$ be the optimal value of MAXIMUM CIRCULATION with k strongly connected components $\{\mathcal{M}_1, \ldots, \mathcal{M}_k\}$. Component $\mathcal{M}_i$ is composed with M_i stations and contributes to a value C_i^* in the optimal MAXIMUM CIRCULATION: $\sum_{i=1}^k C_i^* = Circ^*$.

Let $P_{Circ}^{\vec{N}}$ be the value of the circulation policy with vehicle distribution $\vec{N}$. Let $\vec{N}^*$ be the optimal vehicle distribution for the MAXIMUM CIRCULATION static policy. Let $\vec{N}^U$ be the random uniform vehicle distribution, defined by assigning each of the N vehicles independently to a strongly connected component, with probability $\frac{M_i}{M}$ for component $i \in \{1, \ldots, k\}$.

From Lemma 8, for random uniform vehicle distribution $\vec{N}^U$, the expected uniform stationary distribution $\mathbb{E}[A(\vec{N}^U, M_i)]$ of a vehicle at any station belonging to component i satisfies: $\mathbb{E}[A(\vec{N}^U, M_i)] \geq \frac{N}{N+M-1}$. Therefore:

$$P_{Circ^*}^{\vec{N}^*} \geq \mathbb{E}\left[P_{Circ^*}^{\vec{N}^U}\right] = \mathbb{E}\left[\sum_{i=1}^k A(\vec{N}^U, M_i) C_i^*\right] = \sum_{i=1}^k \mathbb{E}\Big[A(\vec{N}^U, M_i)\Big] C_i^* \geq \frac{N}{N+M-1} Circ^*.$$

Let P_{dyn^*} be the value of an optimum dynamic policy. We have finally:

$$\frac{N}{N+M-1} P_{Dyn^*} \leq \frac{N}{N+M-1} Circ^* \leq P_{Circ^*}^{\vec{N}^*}. \qquad \square$$

Remark 6. *On can deduce from Theorem 9 that: 1) For single strongly component circulation policies, the performance ratio of* MAXIMUM CIRCULATION *is exactly* $\frac{N}{N+M-1}$. *2) For disconnected circulation policies, the performance ratio of the* MAXIMUM CIRCULATION *policy is strictly greater than* $\frac{N}{N+M-1}$ *together with its optimal vehicle distribution and is strictly lower than* $\frac{N}{N+M-1}$ *for the worst deterministic vehicle distribution.*

4.3.3.3 Weak but simple guaranties of performance

We propose now simpler but weaker proofs than the one given in Theorem 9 to prove that MAXIMUM CIRCULATION policy together with its optimal vehicle distribution is an approximation algorithm on dynamic policies.

An exact guaranty for complete demand graphs We first consider the case of complete demand graph. With this property MAXIMUM CIRCULATION contains only one single strongly connected component. Therefore no vehicle distribution has to be specified.

Proposition 8. *For a complete demand graph,* MAXIMUM CIRCULATION *opens all stations and all trips. There is hence only one strongly connected component:* $k = 1$ *and* $\mathcal{M}_1 = \mathcal{M}$.

Proof. Assume there exists a MAXIMUM CIRCULATION λ with value $Circ^*$ with at least two strongly connected components $\mathcal{M}_1$ and $\mathcal{M}_2$. Since the demand graph is complete, choose any $a \in \mathcal{M}_1,\ b \in \mathcal{M}_2$ we have $\Lambda_{a,b} > \lambda_{a,b} = 0$ and $\Lambda_{b,a} > \lambda_{b,a} = 0$. The vector

$$\lambda' = \Big(\big(\lambda_{c,d},\ \forall (c,d) \in \mathcal{D} \setminus \{(a,b),(b,a)\}\big),\ \lambda'_{a,b} = \lambda'_{b,a} = \min\{\Lambda_{a,b}, \Lambda_{b,a}\} \Big)$$

is a circulation with a value $Circ'$ strictly better than the value of the supposed MAXIMUM CIRCULATION λ: $Circ' = Circ^* + 2\min\{\Lambda_{a,b}, \Lambda_{b,a}\}$. □

Proposition 9. *For complete demand graphs,* MAXIMUM CIRCULATION *static policy is a tight* $\frac{N}{N+M-1}$*-approximation on both static and dynamic optimal policies.*

Proof. Let $Circ^*$ be the optimal value of MAXIMUM CIRCULATION and P_{Circ^*} be the value of the static policy provided by MAXIMUM CIRCULATION. For a complete demand graph, MAXIMUM CIRCULATION policy has a single strongly connected component containing the M stations (Proposition 8). From Lemma 1 we have $P_{Circ^*} = \frac{N}{N+M-1} Circ^*$. As shown in Theorem 7, $Circ^*$ is an upper bound on the optimal dynamic policies of value P_{Dyn^*}. Therefore, $P_{Dyn^*} \leq Circ^*$ and

$$\frac{N}{N+M-1} P_{Dyn^*} \leq \frac{N}{N+M-1} Circ^* = P_{Circ^*}.$$

The example of Proposition 7 can be extended to the complete demand graph: Consider a circuit with M stations and maximum demand $\{k, \ldots, k, 1\}$. Complete the maximum demand vector replacing null demands by $\frac{1}{k}$. For any number $M \geq 2$ of stations and any number N of vehicles, the limit (as $k \to \infty$) of the ratio between the value of MAXIMUM CIRCULATION and the generous static policy is $\frac{N}{N+M-1}$. □

A weak guaranty for general demand graphs

Proposition 10. MAXIMUM CIRCULATION *static policy together with its optimal vehicle distribution is a* $\frac{N-M}{N+M}$*-approximation on optimal dynamic policies.*

Proof. Let P_{dyn^*} be the value of a optimal dynamic policy. Let $Circ^*$ be the optimal value of MAXIMUM CIRCULATION with k strongly connected components $\{\mathcal{M}_1, \dots, \mathcal{M}_k\}$. Component $\mathcal{M}_i$ is composed with M_i stations and contributes to a value C_i^* in the MAXIMUM CIRCULATION: $\sum_{i=1}^k C_i^* = Circ^*$. If we allocate N_i vehicles to component $\mathcal{M}_i$, the expected transit is $\frac{N_i}{N_i+M_i-1}C_i^*$ (Lemma 6).

Let $P_{Circ}^{\vec{N}}$ be the value of the MAXIMUM CIRCULATION policy with a vehicle distribution $\vec{N}$ among the k components. Let $\overrightarrow{N^*}$ be the optimal vehicle distribution and let $\overrightarrow{N/M}$ be the vehicle distribution setting $\lfloor \frac{N}{M} M_i \rfloor$ vehicles in each component $\mathcal{M}_i$. We have:

$$P_{Circ^*}^{\overrightarrow{N^*}} \geq P_{Circ^*}^{\overrightarrow{N/M}} = \sum_{i=1}^k \frac{\lfloor \frac{N}{M} M_i \rfloor}{\lfloor \frac{N}{M} M_i \rfloor + M_i - 1} C_i^* \geq \sum_{i=1}^k \frac{\frac{N}{M} M_i - 1}{\frac{N}{M} M_i + M_i} C_i^* = \sum_{i=1}^k \frac{\frac{N}{M} - \frac{1}{M_i}}{\frac{N}{M} + 1} C_i^*.$$

$\forall 1 \leq i \leq k,\ M_i \geq 1$, hence:

$$P_{Circ^*}^{\overrightarrow{N^*}} \geq \frac{N-M}{N+M} \sum_{i=1}^k C_i^* = \frac{N-M}{N+M} Circ^*.$$

Since $Circ^*$ is an upper bound on the optimal dynamic policy with value P_{dyn^*} (Theorem 7), we have:

$$P_{Circ^*}^{\overrightarrow{N^*}} \geq \frac{N-M}{N+M} P_{dyn^*}.$$

□

4.4 Conclusion

We investigated a simpler stochastic VSS pricing problem than the general one presented in Chapter 2. We proposed a heuristic combining MAXIMUM CIRCULATION and a greedy algorithm and studied its performance ratio for the transit maximization. We proved that the provided static policy is a tight $\frac{N}{N+M-1}$-approximation on dynamic and static policies.

Several extensions are natural for this work. We believe that adding transportation times has a minor impact on our results. Moreover, since circulation policies spread vehicles very well among the stations, adding capacities to the stations may still allow these policies to be efficient.

On the other hand, demands that are not stationary over time (such has house-work commute) usually do not benefit from naive steady-state goals: stations in residential areas are better off being full in mornings and empty after work. However, MAXIMUM CIRCULATION heuristics can be generalized to optimize over non-stationary demands, as discussed in Waserhole and Jost (2013b), although no guaranty of performance is provided.

Nevertheless, in dense networks of stations such as Vélib's Paris, some users have flexibilities in their origin and destination stations. The classical (BCMP) queuing network results fall apart under such generalization. Different theoretical tools might be required. Numerical analysis through simulations requires data on the demand. However, the demand is hard to estimate since available data only relate the trips sold and not unsatisfied users.

Chapter 5

Fluid Approximation

Mathematics is the art of giving the same name to different things.

Henri Poincare (1854–1912)

Chapter abstract

An exact measure of the VSS stochastic evaluation model is intractable for real-size systems. To solve the VSS stochastic pricing problem, we hence search for approximations. We present a fluid approximation constructed by replacing stochastic demands with a continuous deterministic flow (keeping the demand rate). The fluid dynamic is deterministic and evolves as a continuous process. The fluid model has for advantage to consider time-varying demand. Solving it with discrete prices seems difficult. For continuous prices, we propose a fluid approximation SCSCLP formulation maximizing the transit. The solution of this program produces a static policy. The optimal value of this SCSCLP is conjectured to be an upper bound on dynamic policies. For stationary demand the fluid model is formulated as a linear program. It produces a static heuristic policy and the value of this LP is proved to be an upper bound on dynamic policies optimization.

Keywords: Fluid approximation; Queuing networks with time-varying demand; Continuous linear program; SCSCLP; Upper bound; s-scaled problem; Piecewise stationary approximation.

Résumé du chapitre

Une mesure exacte du modèle stochastique d'évaluation des systèmes de véhicules en libre service est intractable pour des systèmes de taille réelle. Nous cherchons donc des approximation pour résoudre le problème stochastique tarifaire. Nous présentons une approximation fluide (déterministe) du processus markovien que l'on peut voir comme un problème de plomberie. Le modèle fluide est construit en remplaçant les demandes discrètes stochastiques par des demandes continues déterministes égales aux valeurs des espérances. Les véhicules sont considérés comme un fluide continu, dont la répartition entre les stations évolue de manière déterministe dans un réseau de réservoirs inter-connectés par des tuyaux. Nous montrons qu'optimiser le débit de ce système peut se formuler comme un programme linéaire continu, de type *State Constrained Separated Continuous Linear Program* (SCSCLP), qui peut se résoudre de manière efficace. Cette approximation fluide fournit une politique statique et une borne supérieure sur le problème stochastique de base.

Mots clés : Approximation fluide ; Réseau de files d'attentes avec demandes variant au cours du temps ; Programme linéaire continu ; SCSCLP ; Borne supérieure ; s-scaled problem ; Approximation par morceaux stationnaires.

Contents

This chapter is based on the working paper "Vehicle Sharing Systems pricing regulation: A fluid approximation" (Waserhole and Jost, 2013b).

5.1 Introduction

In this chapter we study a fluid approximation of the VSS stochastic pricing problem in its most general version as defined in Chapter 2.

The fluid model is constructed by replacing the stochastic demands with a continuous flow with the corresponding deterministic rate. It gives a deterministic and continuous dynamic and evolves as a continuous process. Optimizing the fluid model to give heuristics on the stochastic model is a well know technique. It is derived as a limit under a strong-law-of-large numbers, type of scaling, as the potential demand and the capacity grow proportionally large (Gallego and van Ryzin, 1994).

Applications of this principle are available in the literature to deal with revenue management problems, see Maglaras (2006) for instance. However, to the best of our knowledge, there is no direct approach available for a general case including our application, although some papers are theorizing the fluid approximation scheme. Meyn (1997) describes some approaches to the synthesis of optimal policies for multiclass queueing network models based upon the close connection between stability of queueing networks and their associated fluid limit models; Bäuerle (2002) generalizes it to open multiclass queueing networks and routing problems.

To sum up, the fluid model might not be easily constructed and, even if found, the convergence might not be trivial to prove. Sometimes, little modifications called tracking policy have to be made on the solution to be asymptotic optimal or simply feasible. In any case, the fluid approximation is known to give a good approximation and also an upper bound on the optimization gap (Bäuerle, 2000).

In Section 5.2 we present the fluid model for the VSS stochastic pricing problem. We show that the fluid model seems hard to solve for discrete prices but easier for continuous ones. In Section 5.3 we propose a Continuous Linear Program (CLP) for the fluid optimization problem with continuous prices and time-varying demand. This CLP provides a pricing heuristic policy and a conjectured upper bound. In Section 5.4 we restrict to a stationary demand to propose a linear program for the fluid optimization. It can be used to produce heuristic policies for time-varying demand: optimizing independently each time-step where demand is considered stable. In Section 5.5 we discuss the pros and cons of the fluid approximation and formulate some conjectures regarding theoretical aspects.

5.2 The Fluid Model

5.2.1 A plumbing problem

The fluid approximation can be seen as a plumbing problem. Stations are represented by tanks connected by pipes representing the demands. Vehicles are considered as a continuous fluid evolving in this network. The volume of a tank represents the capacity of a station. The length of a pipe represents the transportation time between two stations. The section of a pipe between two tanks a and b represents the demand between stations a and b, it ranges over time from 0 to the maximum demand Λ_t. Figure 5.1 schemes an example with 2 stations. The modeled system has no dynamic interaction with the user. Decisions are static and have to be taken before, once for all. The fluid optimization amount to setting the width of a pipe by changing the price to pass flow in it: the higher the price is, the smaller the pipe (demand) will be.

For a given policy (prices/demands) the deterministic evolution of the system is subject to different constraints. They derive from the first come first served rule that happens in practice. If a pipe (a demand) exists and there is some flow (vehicles) available in the tank (station), the flow has to pass through the pipe. If there is not enough flow to fulfill all pipes (demands), there should be some *departure equity* between them. In other words, the proportion of filling up of all pipes should be equal. However if the arrival tank of a pipe is full, it might be impossible to fulfill this pipe and respect the departure equity as it is. In this case, an *arrival equity* should be applied to all pipes discharging into this tank. In other words, for each pipe, if its discharging tank is full, it has the same proportion of filling up as the other pipes discharging in this tank, otherwise, it has the same proportion of filling up as the other pipes coming from its source tank. We call *equity issues* the problem of respecting the arrival/departure equity to model the evolution of the flow.

5.2.2 Discrete price model

The main goal of this section is to exhibit the complexity of solving the fluid model for discrete prices. Though results of this section might be interesting, the technical aspects are “hard to digest” and not useful to understand the rest of the chapter. The reader only needs to understand the contribution of each section: In Section 5.2.2.1 we propose a non-linear model that formally specifies as a mathematical problem the fluid approximation with discrete price. In Section 5.2.2.2 we show that the fluid dynamic with fixed prices/demands is not linear. Therefore it cannot be model as a linear program and is probably hard to solve.

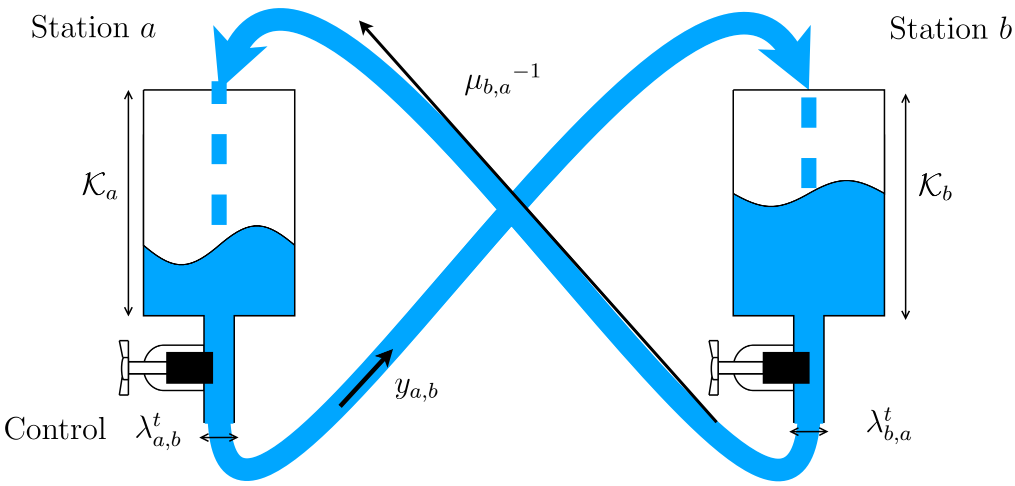


Figure 5.1: A plumbing problem.

5.2.2.1 A non linear model

Before building a mathematical model, we recall the data and define the variables of the model as schemed in Figures 5.2a and 5.2b.

Data:

N	the number of vehicles available;
$\mathcal{M}$	the set of stations;
$\mathcal{K}_a$	the capacity of station a;
$\mathcal{D}$	the set of possible trips ($= \mathcal{M} \times \mathcal{M}$);
$\mu^{-1}_{a,b}$	the transportation time between station a and b
$\lambda_{a,b}(t)$	the demand rate from station a at time t to station b;
$\mathcal{P}_{a,b}(t)$	the set of possible prices to go from station a at time t to station b at time $t + \mu^{-1}_{a,b}$;
$\Lambda(price)$	the function giving the demand for a given price.

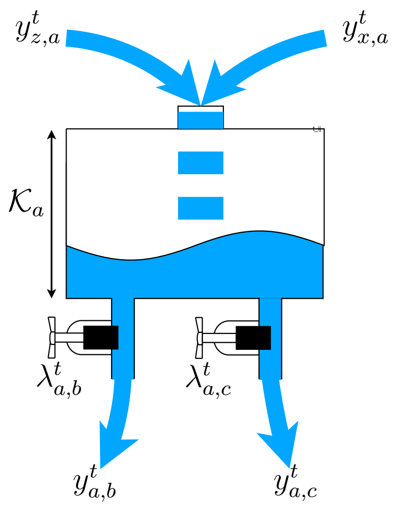
Variables at time t:

$p_{a,b}(t)$	the price to take the trip from station a to station b;
$\phi_a^+(t)$	the proportion of requests accepted among those willing to leave station a;
$\phi_a^-(t)$	the proportion of requests accepted among those willing to arrive at station a that have been accepted to take a vehicle at their departure station;
$y_{a,b}(t)$	the flow leaving station a at time t and arriving at station b at time $t+\mu_{a,b}^{-1}$;
$y_{a,b}^{dep}(t)$	the flow accepted to leave station a but not yet accepted to arrive at station b;
$y_{a,b}^{ref}(t)$	the flow refused by station b "returning to station a" (one has $y_{a,b}^{dep}(t) = y_{a,b}^{ref}(t) + y_{a,b}(t)$), (this variable is not needed and not explicit in the model but helps the understanding) ;
$s_a(t)$	the available stock (number of vehicles) at station a;
$r_a(t)$	the number of parking spots reserved at station a (flow in transit towards a).

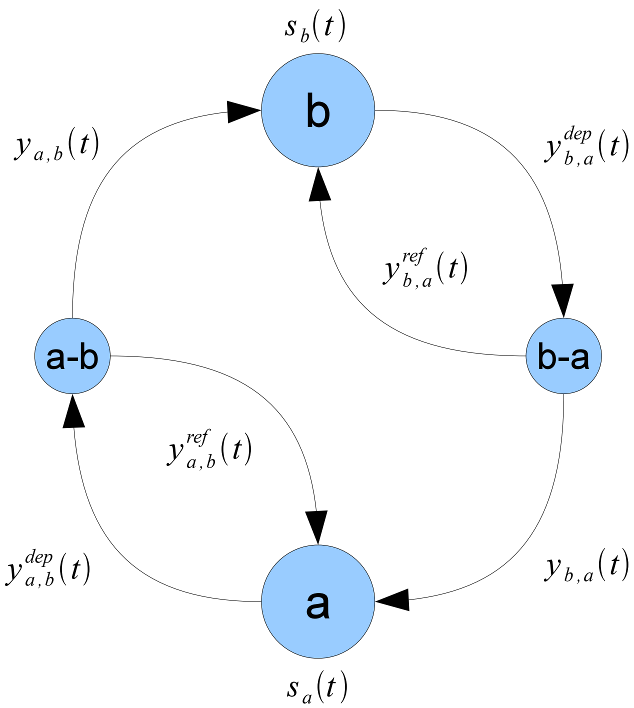
Discrete price model

$$\max \sum_{(a,b)\in D} \int_0^T y_{a,b}(t) \times p_{a,b}(t)\ dt \qquad \text{(Gain)}$$

$$\begin{aligned}
\text{s.t. } & p_{a,b}(t) \in \mathcal{P}_{a,b}(t), && \forall a \in \mathcal{M},\ \forall t \in [0,T], && \text{(Discrete price)}\\
& \lambda_{a,b}(t) = \Lambda(p_{a,b}(t)), && \forall a \in \mathcal{M},\ \forall t \in [0,T], && \text{(Discrete demand)}\\
& \sum_{a\in\mathcal{M}} s_a(0) = N, && && \text{(Flow size)}\\
& s_a(0) = s_a(T), && \forall a \in \mathcal{M}, && \text{(Flow stabilization)}\\
& y_{a,b}^{dep}(t) = y_{a,b}^{ref}(t) + y_{a,b}(t), && \forall (a,b) \in \mathcal{D},\ \forall t \in [0,T], && \text{(Flow conservation)}\\
& s_a(t) = s_a(0) + \int_0^t \sum_b y_{a,b}(\theta) - y_{b,a}(\theta - \mu_{b,a}^{-1})\ d\theta, && \forall a \in \mathcal{M},\ \forall t \in [0,T], && \text{(Flow conservation)}\\
& \phi_a^+(t) = \begin{cases} 1, & \text{if } s_a(t) > 0,\\ \min\left\{1,\ \frac{\sum_b y_{b,a}(t-\mu_{b,a}^{-1}) + y_{a,b}^{ref}(t)}{\sum_b \lambda_{a,b}(t)}\right\}, & \text{otherwise,}\end{cases} && \forall a \in \mathcal{M},\ \forall t \in [0,T], && \text{(Departure equity)}\\
& \phi_a^-(t) = \begin{cases} 1, & \text{if } s_a(t) + r_a(t) < \mathcal{K}_a,\\ \min\left\{1,\ \frac{\sum_b y_{a,b}(t)}{\sum_b y_{b,a}^{dep}(t)}\right\}, & \text{otherwise,}\end{cases} && \forall a \in \mathcal{M},\ \forall t \in [0,T], && \text{(Arrival equity)}\\
& y_{a,b}^{dep}(t) = \phi_a^+(t) \times \lambda_{a,b}(t), && \forall (a,b) \in \mathcal{D},\ \forall t \in [0,T], && \\
& y_{a,b}(t) = \phi_b^-(t) \times y_{a,b}^{dep}(t), && \forall (a,b) \in \mathcal{D},\ \forall t \in [0,T], && \text{(Flow equity)}\\
& r_a(t) = \sum_b \int_0^{\mu_{b,a}^{-1}} y_{b,a}(t-\theta)\ d\theta, && \forall a \in \mathcal{M},\ \forall t \in [0,T], && \text{(Reserved Park Spot)}\\
& s_a(t) + r_a(t) \le \mathcal{K}_a, && \forall a \in \mathcal{M},\ \forall t \in [0,T], && \text{(Station Capacity)}\\
& \lambda_{a,b}(t),\ s_a(t),\ r_a(t),\ y_{a,b}(t) \ge 0, && && \\
& y_{a,b}^{dep}(t),\ y_{a,b}^{ref}(t),\ \phi_a^+(t),\ \phi_a^-(t) \ge 0. && &&
\end{aligned}$$



(a) Incoming and outgoing flows: An equity issue.



(b) Variables for 2 stations.

Figure 5.2: Discrete price fluid model variables.

Remark 7. *It is easy to compute the value of a solution with one price without the flow stabilization constraint. A simple iterative algorithm on the horizon works. With the flow stabilization constraint it is not clear that looping on such iterative algorithm converges to a stable solution.*

5.2.2.2 A non linear dynamic

The previous program might not be the simplest formulation of the discrete price optimization problem. However, as claims the next lemma, the discrete price dynamic is not linear and therefore there exists no linear program modeling the discrete price optimization problem.

Lemma 9. *The fluid model with fixed demands is not linear.*

Proof. A simple evaluation for a given price, hence a given demand λ, presents a non linear dynamic. Figure 5.3 shows an example with integer data where the instantaneous flow is an irrational number. There are 6 stations. At time t, stations a and d are not empty, c and f are not full, b is empty and e is full. All instant demands (λ^t) have for intensity 1. For a matter of simplicity, in the sequel, the time parameter (t) will be implicit. From the paradigm of arrival and departure equity, we deduce the instantaneous value of the flow as follows:

$$
\begin{array}{rrcl}
(b \text{ is empty, no arrival equity}) & y^{dep}_{a,b} = y_{a,b} = \lambda_{a,b} & \rightarrow & y_{a,b} = 1, \\
(\text{Departure equity in } b) & \dfrac{y^{dep}_{b,c}}{\lambda_{b,c}} = \dfrac{y^{dep}_{b,e}}{\lambda_{b,e}} & \rightarrow & y^{dep}_{b,c} = y^{dep}_{b,e} = x, \\
(c \text{ is not full}) & y^{ref}_{b,c} = 0, & & \\
(\text{Flow conservation in } b) & y^{dep}_{b,c} + y^{dep}_{b,e} = y_{a,b} + y^{ref}_{b,d} & \rightarrow & y^{ref}_{b,d} = 2x - 1, \\
(\text{Flow conservation in } b-e) & y^{dep}_{b,e} = y^{ref}_{b,e} + y_{b,e} & \rightarrow & y^{ref}_{b,e} = 1 - x, \\
(\text{Flow conservation in } e) & y_{b,e} + y_{d,e} = 1 & \rightarrow & y_{d,e} = x, \\
(d \text{ is not empty}) & y^{dep}_{d,e} = 1, & & \\
(\text{Arrival equity in } e) & \dfrac{y_{b,e}}{y^{dep}_{b,e}} = \dfrac{y_{d,e}}{y^{dep}_{d,e}} & \rightarrow & x^2 + x - 1 = 0 \leftrightarrow x = \dfrac{-1 \pm \sqrt{5}}{2}.
\end{array}
$$

$\frac{-1-\sqrt{5}}{2} < 0$, therefore $y_{b,e} = \frac{-1+\sqrt{5}}{2}$ which is an irrational number. Since all numbers in the data are rational, the flow dynamic for a given price is not linear. □

5.2.3 Continuous price model

We can avoid dealing with equity issues for the continuous prices fluid evaluation. The trick is to *always fulfill the pipes*, in other words to have a flow y between two stations that is exactly equal to the demand λ for taking this trip. It is always possible when assuming a continuous elastic demand, *i.e.* there exists a price $p(\lambda^t_{a,b})$ to obtain any demand $\lambda^t_{a,b} \in [0, \Lambda^t_{a,b}]$. Solutions respecting this trick define the solution space of the fluid model with continuous price. More formally the "fluidification" of the state space is the following:

Continuous price fluid model solution space

- A continuous space replaces the discrete one:

$$
\mathcal{S}^F = \left\{ \Big(n_a \in \mathbb{R} : \ a \in \mathcal{M}, \ n_{a,b} \in \mathbb{R} : \ (a,b) \in \mathcal{D}, \ t \in [0,T] \Big) \right.
$$
$$
\left. / \sum_{i \in \mathcal{M} \cup \mathcal{D}} n_i = N \ \& \ n_a + \sum_{b \in \mathcal{M}} n_{b,a} \leq \mathcal{K}_a, \ \forall a \in \mathcal{M}, \ \forall t \in [0,T] \right\}
$$

- A continuous deterministic demand with rate $\lambda^t_{a,b}$ replaces the discrete stochastic one.
- A deterministic transportation time of duration ${\mu^t_{a,b}}^{-1}$ replaces the stochastic one.

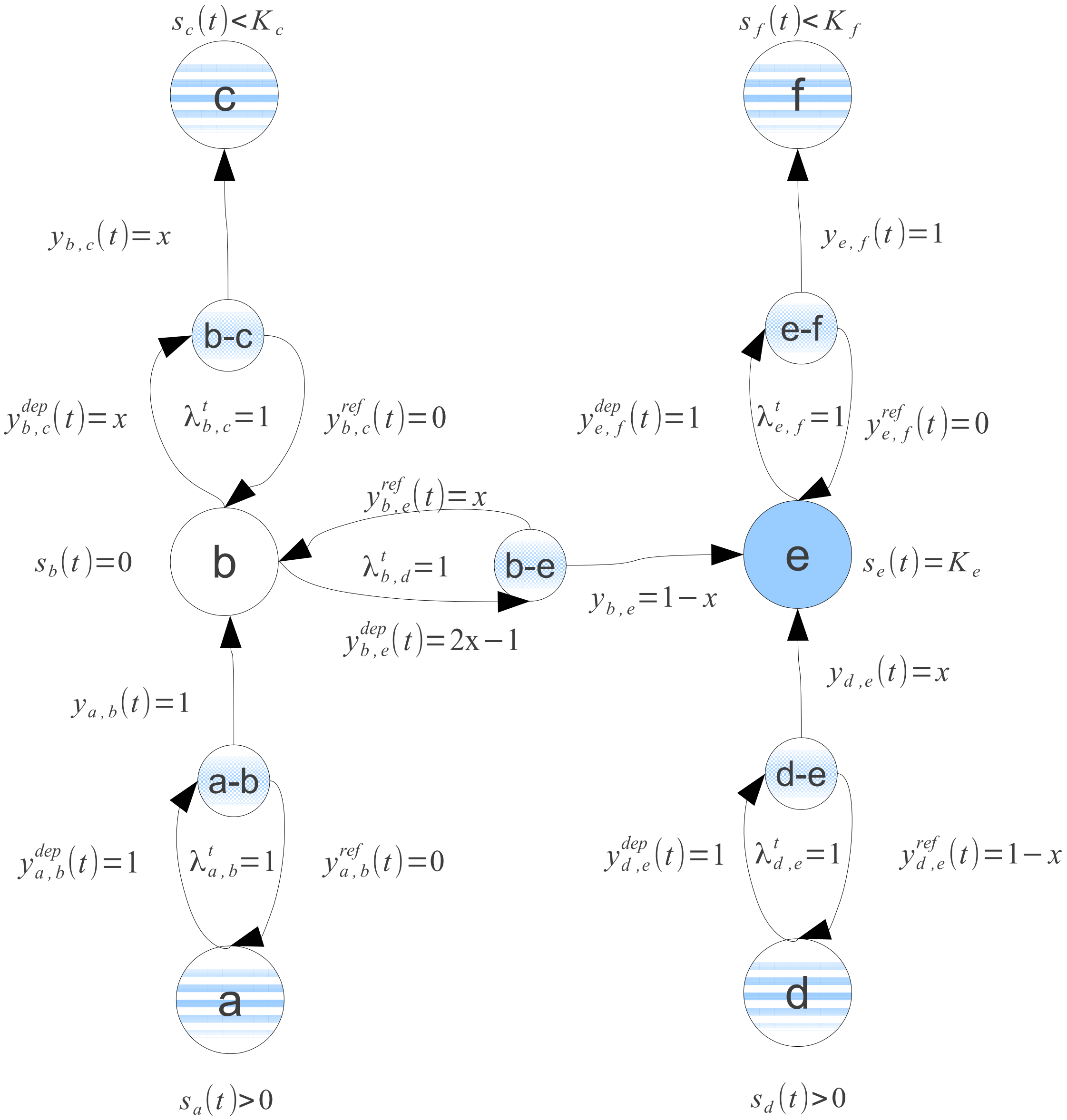


Figure 5.3: Discrete price fluid dynamic is non linear. Exhibition of an irrational solution, here $x = \frac{-1\pm\sqrt{5}}{2}$.

- λ is a periodic flow over time with capacity constraints $\lambda^t_{a,b} \in [0, \Lambda^t_{a,b}]$.

5.3 Time-varying demand CLP formulation

In this section we formalize the fluid approximation of the VSS stochastic continuous pricing problem as a continuous program.

5.3.1 Continuous linear programming literature review

Continuous Linear Programs (CLP) are introduced by Bellman (1953). Although many studies have been made on general CLP, they remain difficult to solve exactly (Anderson and Nash, 1987). Recently Bampou and Kuhn (2012) propose a generic approximation scheme for CLP, where they approximate the policies by polynomial and piecewise polynomial decision rules. Fluid relaxations are a specially structured class of CLP called State Constrained Separated Continuous Linear Programs (SCSCLP). Luo and Bertsimas (1999) introduce SCSCLP, establish strong duality, and propose a convergent class of algorithms for this problem. Their algorithm is based on time discretization and removes redundant breakpoints but solves quadratic programs in intermediate steps. The complexity of solving SCSCLP is still open. In fact, the size of the optimal solutions may be exponential in the input size. In the absence of upper bounds on storage [1], SCSCLP are called Separated Continuous Linear Programs (SCLP). Anderson *et al.* (1983) characterize extreme point solutions of SCLP. For problems with linear data, they show the existence of an optimal solution in which the flow-rate functions are piecewise constant with a finite number of pieces. Weiss (2008) presents an algorithm which solves SCLP in a finite number of steps, using an analog of the simplex method. Fleischer and Sethurama (2005) provide a polynomial time algorithm with a provable approximation guaranty for SCLP.

5.3.2 A continuous linear solution space

If at any time $t \in [0, T]$ and for any trip $(a, b) \in \mathcal{D}$, the demand $\lambda_{a,b}(t)$ is exactly equal to the instantaneous flow $y^t_{a,b}$ passing between two stations a and b, the continuous price fluid model solution space can be expressed as a Continuous Linear Program (CLP). For all time $t \in [0, T]$, we define the following variables:

$\lambda_{a,b}(t)$	is the demand to go from station a to station b at time $t + \mu^{-1}_{a,b}$ (with price $p(\lambda_{a,b}(t))$);
$s_a(t)$	is the available stock of vehicles at station a;
$r_a(t)$	is the number of parking spots reserved at station a.

1. In our application it is the case when considering stations with infinite capacities

Fluid Solution Space (5.1)

$$\sum_{a\in\mathcal{M}} s_a(0) = N, \qquad \text{(Flow size)} \tag{5.1a}$$

$$s_a(0) = s_a(T), \qquad \forall a \in \mathcal{M}, \qquad \text{(Flow stabilization)} \tag{5.1b}$$

$$s_a(t) = s_a(0) + \int_0^t \sum_{(b,a)\in\mathcal{D}} \lambda_{b,a}(\theta - \mu_{b,a}^{-1}) - \lambda_{a,b}(\theta)\, d\theta, \qquad \forall a \in \mathcal{M},\ \forall t \in [0,T], \qquad \text{(Flow conservation)} \tag{5.1c}$$

$$0 \le \lambda_{a,b}(t) \le \Lambda_{a,b}^t, \qquad \forall a,b \in \mathcal{M},\ \forall t \in [0,T], \qquad \text{(Maximum demand)} \tag{5.1d}$$

$$r_a(t) = \sum_{b\in\mathcal{M}} \int_0^{\mu_{b,a}^{-1}} \lambda_{b,a}(t-\theta)\, d\theta, \qquad \forall a \in \mathcal{M},\ \forall t \in [0,T], \qquad \text{(Reserved park spot)} \tag{5.1e}$$

$$s_a(t) + r_a(t) \le \mathcal{K}_a, \qquad \forall a \in \mathcal{M},\ \forall t \in [0,T], \qquad \text{(Station capacity)} \tag{5.1f}$$

$$s_a(t) \ge 0, \quad r_a(t) \ge 0, \qquad \forall a \in \mathcal{M},\ \forall t \in [0,T]. \tag{5.1g}$$

Equation (5.1a) defines the amount of flow to be equal to the N vehicles available. Equations (5.1b) constrain the flow to be stable, *i.e.* cyclic over the horizon. Equations (5.1c) are a continuous version of the classic flow conservation. Equations (5.1d) constrain the flow on a demand edge to be less or equal than the maximum demand. Equations (5.1e) set the reserved parking spot variable. Equations (5.1f) constrain the maximum capacity on a station and the parking spot reservation: For a station the number of reserved parking spots plus the number of vehicles already parked should not exceed its capacity.

This model assumes that there is an "off period" between the cycling horizons where all vehicles are parked at a station. If it is not the case, only some small changes have to be made in the flow equations.

If the static policies provided has a connection graph $G(\mathcal{M}, \int_0^T \lambda)$ with several strongly connected components, the vehicle distribution among the station is set according to vector $s(0)$.

5.3.3 A SCSCLP for transit maximization

Maximizing the number of trips sold amounts to maximizing the expected transit of the system: $\sum_{(a,b)\in\mathcal{D}} \int_0^T \lambda_{a,b}(t)\, dt$. This objective is linear and together with the Fluid Solution Space (5.1), it defines a State Constrained Separated Continuous Linear Programs (SCSCLP) solving the continuous price fluid model policy maximizing the transit.

Transit maximization – Fluid SCSCLP (5.2)

$$\max \sum_{(a,b)\in\mathcal{D}} \int_0^T \lambda_{a,b}(t)\ dt \qquad \text{(Transit)} \tag{5.2a}$$

$$\text{s.t. } (5.1a) - (5.1g). \qquad \text{(Fluid Solution Space)} \tag{5.2b}$$

5.3.4 A non linear continuous program for revenue optimization

Even if maximizing the expected revenue of a VSS system is not in the scope of this study, we propose the following continuous non linear program to formally define the problem.

Fluid for revenue maximization – Continuous non linear program

$$\max \sum_{(a,b)\in D} \int_0^T \lambda_{a,b}(t) \times price(\lambda_{a,b}(t))\ dt \qquad \text{(Gain)}$$

$$\text{s.t. } (5.1a) - (5.1g). \qquad \text{(Fluid Solution Space)}$$

In this formulation the gain computation is not linear. If the gain function $g^t_{a,b}(\lambda^t_{a,b}) = \lambda^t_{a,b} \times p(\lambda^t_{a,b})$ is concave [2], it amounts to minimizing a convex function for which there exists efficient solutions methods [3].

5.4 Stationary demand LP formulation

When we consider a stationary demand ($\lambda^t = \lambda$), the steady-state fluid model can be reduced to the following LP (5.3).

2. In particular $p(\lambda) = \lambda^{-\alpha}$ with $\alpha \in [0, 1]$ is a concave function.
3. For instance it is possible to make a linear approximation of the gain to obtain an approximate SCSCLP maximizing the revenue of the system.

Stable fluid LP (5.3)

$$\max \sum_{(a,b)\in\mathcal{D}} \lambda_{a,b} \qquad \text{(Transit)} \tag{5.3a}$$

$$\text{s.t.} \sum_{(a,b)\in\mathcal{D}} \lambda_{a,b} = \sum_{(b,a)\in\mathcal{D}} \lambda_{b,a}, \qquad \forall a \in \mathcal{M}, \qquad \text{(Flow conservation)} \tag{5.3b}$$

$$0 \leq \lambda_{a,b} \leq \Lambda_{a,b}, \qquad \forall (a,b) \in \mathcal{D}, \qquad \text{(Maximum Demand)} \tag{5.3c}$$

$$\sum_{(a,b)\in\mathcal{D}} \frac{1}{\mu_{a,b}}\lambda_{a,b} \leq N, \qquad \text{(Vehicles number)} \tag{5.3d}$$

$$\sum_{b\in\mathcal{M}} \frac{1}{\mu_{a,b}}\lambda_{a,b} \leq \mathcal{K}_a, \qquad \forall a \in \mathcal{M}. \qquad \text{(Station capacities)} \tag{5.3e}$$

The objective function (5.3a) maximizes the throughput. Equations (5.3b) conserve the flow. Equations (5.3c) constrain the maximal demand on each trip. Equation (5.3d) constrains the number of vehicles in the system according to Little's law. Equations (5.3e) constrain the reservation of parking spot with respect to the station capacity.

In order to get a better understanding of stable fluid LP (5.3), consider an equivalent formulation, more natural, that explicitly specifies where the N vehicles are. A vehicle can be either in a station, represented by variables $s_a > 0$, or in transit between two stations represented by variable $y_{a,b}$. Since the demand is continuous and deterministic, for any trip $(a,b) \in \mathcal{D}$ and at any instant, $y_{a,b}\frac{1}{\mu_{a,b}}\lambda_{a,b}$. Hence, the number of vehicles in the system and the parking spot reservation constraints can then be represented by the following equations:

$$\sum_{(a,b)\in\mathcal{D}} \frac{1}{\mu_{a,b}}\lambda_{a,b} + \sum_{a\in\mathcal{M}} s_a = N,$$

$$\sum_{b\in\mathcal{M}} \frac{1}{\mu_{a,b}}\lambda_{a,b} + s_a \leq \mathcal{K}_a, \qquad \forall a \in \mathcal{M}.$$

If there are less vehicles than the total number of parking spots, *i.e.* $N \leq \sum_{a\in\mathcal{M}} \mathcal{K}_a$, these equations define the same space as Equations (5.3d) and (5.3e).

Theorem 10. *For a stationary demand, the optimal objective value of stable fluid LP (5.3) provides an upper bound on dynamic policies.*

Proof. From any dynamic policy, with transition rate $\lambda^s_{a,b} \leq \Lambda_{a,b}$ in state $s \in \mathcal{S}$ for trip $(a,b) \in \mathcal{D}$, we construct a stable fluid LP (5.3) solution with same value. Let e_a be the unit vector for component $a \in \mathcal{M}$: $e_a = (0,\ldots,0,n_a = 1,0,\ldots,0)$.

Under dynamic policy λ, the stationary distribution π over the state space $\mathcal{S}$ of the continuous-time Markov chain defined by λ satisfies:

$$\begin{aligned} & \sum_{s\in\mathcal{S}} \pi_s = 1, & \\ \sum_{\substack{(a,b)\in\mathcal{D}\\ s-e_a+e_b\in\mathcal{S}}} \pi_s\lambda^s_{a,b} = & \sum_{\substack{(b,a)\in\mathcal{D},\ s'\in\mathcal{S}\\ s'-e_b+e_a=s}} \pi_{s'}\lambda^{s'}_{b,a}, & \forall s\in\mathcal{S}, \\ & \pi_s \geq 0, & \forall s\in\mathcal{S}. \end{aligned}$$

Let $\lambda'_{a,b}$ be the expected transit for any trip $(a,b)\in\mathcal{D}$: $\lambda'_{a,b} = \sum_{s\in\mathcal{S}} \pi_s\lambda^s_{a,b}$. We show that λ' is a stable fluid LP (5.3) solution: Flow conservation constraints (5.3b) are satisfied because in the steady state of a dynamic policy, a station receives as many vehicles as it is sending. The maximum demand constraints (5.3c) are satisfied since $\sum_{s\in\mathcal{S}} \pi_s = 1$ and hence:

$$\lambda'_{a,b} = \sum_{s\in\mathcal{S}} \pi_s\lambda^s_{a,b} \leq \sum_{s\in\mathcal{S}} \pi_s\Lambda_{a,b} = \Lambda_{a,b}, \quad \forall (a,b)\in\mathcal{D}.$$

The vehicles number constraints (5.3d) and the reservation of parking spots constraints (5.3e) are trivially respected in the continuous-time Markov chain.

Finally, the expected transit of the system is equal to $\sum_{(a,b)\in\mathcal{D}} \lambda'_{a,b}$ which is the value of stable fluid LP (5.3) solution λ'. $\square$

Remark 8. *For infinite capacities, when the number of vehicles tends to infinity, stable fluid LP* (5.3) *amounts to solving a* Maximum Circulation *problem. In Chapter 3 we showed that* Maximum Circulation *static policy provides the best dynamic policy when the number of vehicles tends to infinity.*

Stable fluid PSA Stable fluid LP (5.3) can be used to produce static policies for time-varying demand with a Pointwise Stationary Approximation (PSA) (Green and Kolesar, 1991). It amounts to solving a stable fluid LP (5.3) on each time step where the demand is considered stationary. We name this heuristic *stable fluid PSA* (5.3). This sum on each time step of stable fluid LP (5.3) value does not provide an upper bound anymore. However, this heuristic policy is easy to compute.

5.5 Discussion

5.5.1 Advantages/Drawbacks of fluid approaches

The main advantage of the fluid SCSCLP (5.2) is to consider time-dependent demands, thus providing a macro management of the tide phenomenon. The policies produced are static but the fluid model may also help designing dynamic ones (Maglaras and Meissner, 2006) with a multiple start heuristic for instance.

A weakness of this approach is that there is no control on the static policy time step. Indeed, the optimal solution might lead to change the prices every 5 minutes which seems not suitable in practice. Moreover, since it is a deterministic approximation, this model does not take into account the stochastic aspect of the demand. We suspect that it can be a problem for systems with low demands where the variance is then higher, or for systems with small station capacities where problematic states (empty or full) are more frequent.

5.5.2 Questions & Conjectures

Fluid model as an asymptotic limit To the best of our understanding, Fluid Solution Space (5.1) is a fluid approximation of the VSS stochastic problem. In the literature, *e.g.* Maglaras (2006), it is classic to interpret this model as an asymptotic limit of a s-scaled problem sequence.

s-scaled stochastic continuous pricing problem

- The stochastic process evolves in a scaled discrete space:

$$\mathcal{S}(s) = \left\{ \left(n_a \in \frac{\mathbb{N}}{s} : \ a \in \mathcal{M}, \ n^r_{a,b} \in \frac{\mathbb{N}}{s} : \ ((a,b), r) \in \mathcal{D} \times R, \ t \in \frac{\mathcal{T}}{s} \right) \right.$$
$$\left. / \sum_{i \in \mathcal{M} \cup \mathcal{D} \times R} n_i = N \ \& \ n_a + \sum_{r \in R} \sum_{b \in \mathcal{M}} n^r_{b,a} \leq \mathcal{K}_a, \ \forall a \in \mathcal{M}, \ \forall t \in \frac{\mathcal{T}}{s} \right\},$$

  with $R := \{1, \ldots, s\}$ and for any set X: $\frac{X}{s} = \{\frac{1}{s}, \frac{2}{s}, \ldots, |X|\}$.
- The state space contains fractions instead of integers and the basic unit corresponding to a vehicle (job) and a time step is $1/s$.
- Each time step $t \in \mathcal{T}$ is divided into s parts with duration $\tau^t s^{-1}$.
- The route/transportation time from station a to station b is represented by s servers in series with rate $\mu^{t,r}_{a,b}(s) = s\mu^t_{a,b}$.
- The maximum time-varying transition rates are accelerated by a factor s: $\Lambda^t_{a,b}(s) = s\Lambda^t_{a,b}$.

→ A solution is a continuous control on the prices for each trip, at each time step. Any demand $\lambda^t_{a,b}(s) \in [0, \Lambda^t_{a,b}(s)]$ can be obtained at a price $p^t_{a,b}(s) = \frac{1}{s} p^t_{a,b}(\frac{1}{s}\lambda^t_{a,b}(s))$.

The above scaling allows the convergence of not only the rewards, but also of the state process. We do not include a mathematical study of the convergence model to the fluid model, this is beyond our scope. However, in simulation (Section 6.5.3), Fluid SCSCLP (5.2) seems to converge towards the s-scaled problem.

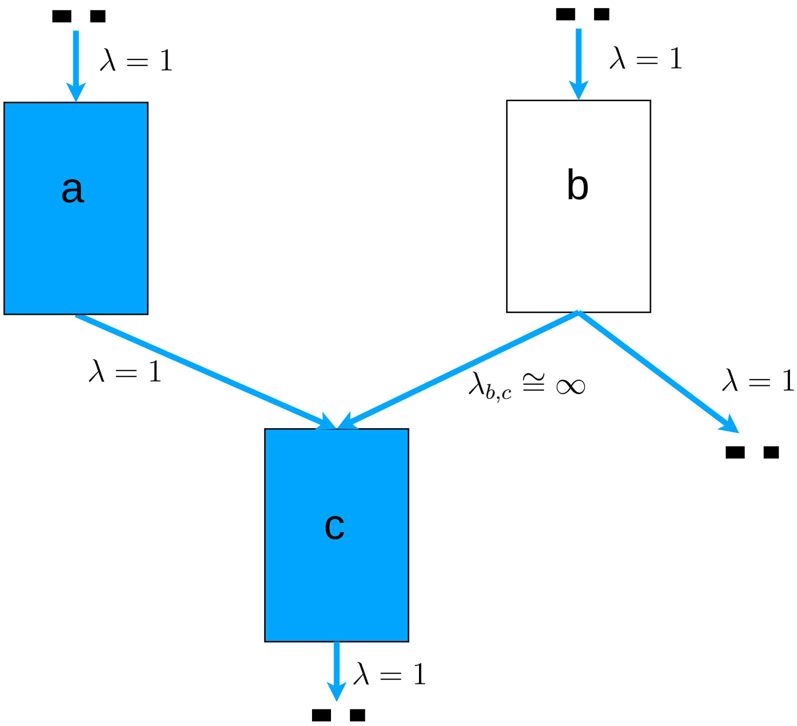


Figure 5.4: Counter example for the convergence of the s-scaled problem as $s \to \infty$ towards the discrete-price fluid model.

Conjecture 1. *Static optimal policies (and their values) of the s-scaled problem converge towards optimal policies of Fluid SCSCLP* (5.2) *when $s \to \infty$.*

Remark 9. *For discrete prices/controls, the s-scaled problem does not converge as s tends to infinity to the discrete-price fluid model (given in Section 5.2.2.1). Indeed, as shows the example of Figure 5.4*[4]*, the evaluation for a given price of the s-scaled problem does not converge to the discrete fluid model. In this instance, the difference is exhibited when the demand for the trip between stations b and c tends to infinity. In the discrete-price fluid model, given a fixed demand λ, the flow $y_{b,c}$ from station b to station c tends to 1 as $\lambda_{b,c}$ tends to infinity. While this value differs for the s-scaled problem evaluation where it is equal to:* $\lim_{s\to\infty} \lim_{\lambda_{b,c}\to\infty} y_{b,c} = \frac{2}{3}$. *However this is not a counter example to Conjecture 1.*

Fluid model as an upper bound One would expect that the uncertainty in sales in the stochastic problem results in lower expected revenues than in the deterministic one. It is shown in many applications as in Gallego and van Ryzin (1994). However

4. We are greatful to Nicolas Gast for pointing out this problem and providing this counter example.

for our problem, we have only been able to prove that the fluid optimal value function gives an upper bound for stationary demands (Theorem 10).

Conjecture 2. *The value of Fluid SCSCLP* (5.2) *optimal solution is an upper bound on dynamic policies of the s-scaled problem ($\forall s$).*

Complexity of the fluid approximation The complexity of solving the fluid approximation is open. For a stationary demand and finite station capacities, the fluid approximation for the VSS discrete pricing stochastic problem seems "hard" to compute since its dynamic is non linear for a single discrete price. For a stationary demand, the fluid approximation for the VSS stochastic continuous pricing transit maximization problem is polynomially solvable in the number of stations M and constant in the number of vehicles N. Indeed Stable fluid LP (5.3) gives the optimal policy (that is fully static) solving this problem.

For time-varying demands, the fluid model optimal static policy, solution of Fluid SCSCLP (5.2), may have an exponential number of "price patterns" in the network size (Fleischer and Sethurama, 2005). To strike this explosion, we might restrict our research to fully static policies, where prices do not depend on time. Fully static policies have a compact formulation but they are not dominant among general static policies. Moreover, solving the fluid approximation for the VSS fully static trip/station policies transit maximization problem is APX-hard for time-varying demands. The proof can be done with the same complexity argument as given for the deterministic VSS trip/station pricing problem, arising in the scenario-based approach in Chapter 3.

Chapter 6

Simulation

Measure what is measurable,
and make measurable what is
not so.

Galileo Galilei (1564–1642)

Chapter abstract

We want to estimate the potential impact of pricing in VSS. In the previous chapters we have formulated heuristic policies and upper bounds. We test them on case studies. A practical case study is conduced on Capital Bikeshare historical data. A simple demand pattern is generated from these data. We show that for such demand there is no potential gain for pricing policies. It exhibits the problem of accessing the real demand. A simple reproducible benchmark and an experimental protocol is proposed. We exhibit that the pricing leverage needs to be considered jointly with the best fleet sizing. The static fluid heuristic policy appears the best one on the simulations. It allows to increase between 10% to 30% the number of trips sold. Max Flow With Reservation provides the best upper bound. Optimization gaps for dynamic policies optimization are between 50% to 100%.

Keywords: Monte-Carlo simulation; Benchmark; Experimental protocol; Pricing; Fleet sizing; SCSCLP time discretization; Optimization gap; Real-case analysis; Reservation constraint.

Résumé du chapitre

Nous voulons estimer l'impact potentiel des politiques tarifaires dans les systèmes de véhicules en libre service. Dans les chapitres précédents nous avons proposé différentes politiques heuristiques ainsi que des bornes supérieures sur les gains possibles d'optimisation. Nous effectuons des tests sur des cas d'études. Un cas d'étude réel est analysé sur les données d'exploitation de Capital Bikeshare. Un patron de demande simple est extrapolé. Nous montrons que pour une telle demande il n'y a pas de gain d'optimisation. Cela met en exergue la nécessité d'accéder à la demande réelle. Un benchmark simple et reproductible ainsi qu'un protocole expérimental sont proposés. Nous montrons que l'étude des politiques tarifaires doit se faire conjointement avec un dimensionnement optimal de la flotte de véhicules. La politique statique donné par l'approximation fluide apparait être la meilleure dans nos simulations. Elle permet de d'améliorer entre 10% à 30% le nombre de trajets vendus. Flot Max Avec Réservation fournit la meilleure borne supérieure. Des gains d'optimisations de l'ordre de 50% à 100% sont observés pour les politiques dynamiques.

Mots clés : Simulation de Monte-Carlo ; Benchmark ; Protocole expérimental ; Tarification ; Dimensionnement de flotte ; Discrétisation temporelle d'un SCSCLP ; Gain potentiel d'optimisation ; Analyse de cas réel ; Contrainte de réservation.

Contents

Part of this chapter is based on the working paper "Vehicle Sharing Systems pricing regulation: A fluid approximation" (Waserhole and Jost, 2013b).

6.1 Introduction

6.1.1 How to estimate pricing interest?

To estimate the impact of pricing in VSS, we need to test our pricing policies and upper bounds on case studies. Our models are based on the VSS stochastic evaluation model defined in Chapter 2 that considers a simple real-time station-to-station reservation protocol and a continuous elastic demand. A case study is hence an instance of a city that defines a set of stations with their capacities, a distance matrix and the maximum time-dependent demand per trip. The number of vehicles available is not fixed since it is an important leverage of optimization (see Section 6.3.3).

We compare the different strategies with the VSS stochastic evaluation model [1]. However, since measuring it exactly is intractable, we estimate its value through Monte-Carlo simulation.

6.1.2 Instance generation – Literature review

A benchmark is a set of case studies/instances. To the best of our knowledge, no benchmark exists in the VSS literature even though some simulation analyses have already been conduced. We characterize three different approaches regarding the instances generation:

- *Random instances* that are easy to generate but for which optimization results are hard to interpret. For instance Chemla *et al.* (2013) generate random instances with a stationary demand.

1. We could have also produced heuristic policies with a simple model and then tested them in a more complex one. For instance, in our case we could neglect the time flexibility in the solution model but consider it in the simulation.

- *Real-data inspired instances* that have some kind of aura because of their real-world origin, even thought they can be corrupted, too specific and not relevant for general interpretations. For instance Pfrommer *et al.* (2013) generate instances based on Barclays Cycle Hire data. They assume 100% service rate for departure in the historical data. Potential customers who could not rent a bicycle due to an empty station are excluded, as they are not recorded in the historical data. They somewhat justify this assumption by the considerable repositioning effort made by the operator of Barclays Cycle Hire BSS.
- *Toy instances* that are simple on purpose to be easier to interpret. For instance Fricker *et al.* (2012) consider a stationary demand and model the demand heterogeneity through clusters of stations having the same behavior. They conduce simulations with a stationary demand and two types of stations, *i.e.* only two values for demand Λ.

6.1.3 The demand estimation problem

Contrary to Pfrommer *et al.* (2013), we doubt that most of the demand is captured in the historical data. At least one needs to consider the censored demand, *i.e.* demand of unserved users that showed up but have been unable to take a trip. Rudloff *et al.* (2013) tackle this problem, they intend to estimate the original (uncensored) demand for bikes and parking spots on Citybike Wien historical data. The estimated station-demand is useful for redistribution of bicycles in a bike-sharing system. We need a demand per trip for the pricing optimization. Unfortunately, according to Rudolff[2], extending their method to characterize the probabilistic distribution for each trip is out of reach with the current computational capacity.

Moreover, we suspect that rebuilding this demand only with historical data might not be relevant because users are learning from the system. Indeed, if three times in a row a user is stuck with a vehicle in an area without any free parking spot, he will probably never take this trip again and will be hidden from the system point of view (he is not part of the censored demand anymore). Moreover, with new types of protocols, such as parking spot reservation, a new demand might be created. To sum up, we think that historical data can be used for balancing strategy optimization, but not for pricing strategy since incentive policies count on using current unserved demand (intuition corroborated in practice, see Section 6.2).

2. Informal communication in Rome at EURO 2013 conference.

6.1.4 Plan of the chapter

In Section 6.2, a real case study is investigated on Capital Bikeshare historical data. It illustrates the importance of considering a real demand and not only using directly historical data. Since the real demand is not accessible, and moreover to isolate and understand the phenomenons at stake, a simple reproducible benchmark (with toy instances) is proposed in Section 6.3. It intends to capture demand intensity, gravitation and tide influences. We explain how to size the instances in order to have reasonable values. In Section 6.4, we compare by simulation on the simple benchmark the pricing strategies presented in the previous chapters. We show that pricing seems to be a relevant leverage and exhibit optimization gaps. In Section 6.5, we investigate some technical aspects regarding the algorithm implementations. We show that solving optimally the fluid model does not provide the best heuristic policy. The influence of the reservation constraint on computation time and quality is studied. The conjecture regarding the convergence of a s-scaled problem toward the fluid model is experimentally tested.

6.2 A real-case analysis

6.2.1 A trivial demand generation

Capital Bikeshare BSS in Washington D.C. provides a free access to its historical data on its website. We use the trips sold from the first quarter of 2013 to create an instance on a week horizon. We assume that all the demand is contained in the data (as in Pfrommer *et al.* (2013)). The demand is considered piecewise stationary on 60 minutes length time steps. Each hour, the stochastic time-varying arrival rate per trip is to the average demand for this hour in the data.

The real system contains about 200 stations and 1800 bikes available. We simulate it with 200 stations with uniform capacity 20. Figure 6.1 compares the performance of the generous policy, the fluid heuristic and the fluid upper bound. The generous policy sells about 3000 trips per week for a 45% vehicle proportion (≈1800 bikes). The optimal number of trips sold is about 4000/week and is attained with 80% vehicle proportion.

Regarding optimization, the fluid heuristic upper bound indicates that there is almost no gap for dynamic pricing optimization. Indeed the generous policy and the upper bound curves are almost identical. Something surprising is that the fluid heuristic is decreasing dramatically the number of trips sold.

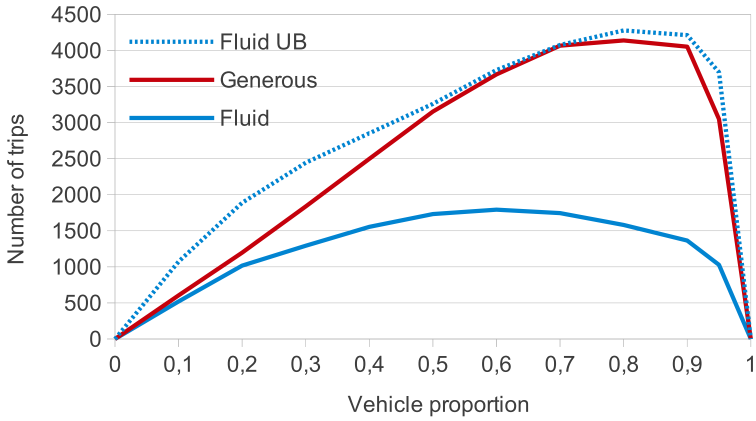


Figure 6.1: Capital Bikeshare case study.

6.2.2 Discussion

In the data, 30 000 trips are sold per week in average. However, in the simulation the generous policy is only able to sell at most 4000 trips. We explain this difference as follows.

In the simulation we do not use bike redistribution contrary to the real context. Without this regulation, the unbalanced demand in the city drives the system quickly into a poor state. Indeed as Figure 6.2 shows, there is only a third of the stations that have a demand for bikes and parking spots relatively balanced. The two other thirds have either a bike or a parking spot deficit. We considered a uniform station capacity that is not the case in reality. Station sizing might be a leverage to prevent the system from being too unbalanced. We used a reservation protocol without any spatial/temporal flexibility. The trip requests arrive randomly, and not structured as it was the case in the original scenario.

The poor performance of the fluid heuristic might be due to the low demand intensity. Indeed 30 000 trips per week is roughly equal to a demand of 2.25 trips per bike per day, or of 1.2 (outgoing) trips per hour per station (considering days of 18 hours). As we will explain in Section 6.5.1, for low demand the variance around the average is high and the fluid deterministic approximation is then unable to cope with randomness. At this stage we notice that Capital Bikeshare (2010) has a relatively low utilization [3]. In comparison, most other schemes report usage rates of around 3–6 trips per bike per day (Fishman *et al.*, 2012).

Finally, regarding the lack of gap for pricing optimization, we think that it is due to the fact that we consider trips sold historical data and not the real demand.

3. We have taken winter trips, the system is more used in summer but nothing dramatic.

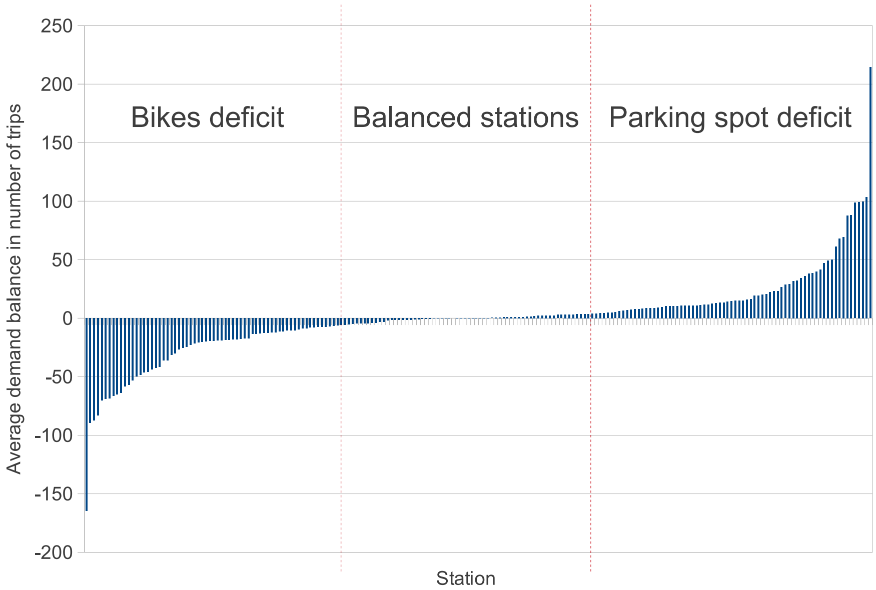


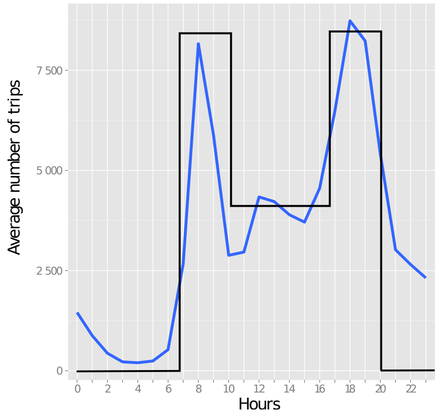
Figure 6.2: Station average demand balance on a week horizon.

Indeed, the trips sold form a type of spatio-temporal flow. In fact any pricing policy implies its spatio-temporal flow, serving only part of the demand. Therefore, in the historical data, there are no alternatives possible to the "original" flow. We think that if the real demand was not hidden, we would have more leverages for a better management of which trips to serve. We should hence pay attention to the necessity of testing the optimization leverage on uncensored (real/potential) demand, in order to be objective when measuring their interest.

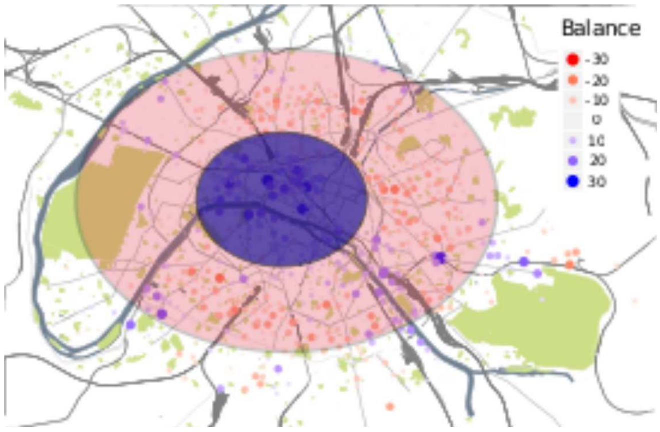
6.3 A simple reproducible benchmark

6.3.1 Origin

We recall part of the discussion regarding system utilization of Section 1.2.3 page 21. In the literature, many data-mining studies have been done on BSS. Their goal is to find groups of stations with similar temporal usage profiles (incoming and outgoing activity/hour) taking into account the week-days /week-end discrepancy. They usually report the same phenomenon: there are roughly two day patterns, a week day and a week-end day. Côme (2012) studies Vélib' historical data. Figure 6.3a represents the average number of trips sold along a week day in Vélib'. It has the two rush hour peaks corresponding to a morning and an evening commute. Figure 6.3b represents the bike balance at Vélib' stations in the morning. Remark the separation into two types of stations: those with a clear positive



(a) A week day. The tide is approximated by a piecewise stationary demand.



(b) Spatial distribution of morning tide: approximation by two types of station.

Figure 6.3: Utilization of Vélib' trip historical data to specify a simple benchmark. Source Côme (2012).

and those with a clear negative balance. This imbalance is the result of one of the spatio-temporal clusters identified by Côme (2012), that he characterizes as a "house-work" demand. Together with the "evening opposite flow", the "work-home" cluster, we name this spatio-temporal phenomenon *tide*. Côme (2012) exhibits in total five clusters: house-work, lunch, work-house, evening and spare time. We use these analyses to specify a benchmark.

6.3.2 Instances

We recall that the following instances are toys, they do not intend to be exhaustive and capture all VSS dynamic specificities. Nevertheless, they have the advantage to be simple, reproducible and we hope they help to characterize interesting phenomenon.

A city formed with stations on a grid We consider a VSS implemented in a city where stations are positioned on a grid of width w, length l and travel time unity $t_{\min} = 15$ (closest distance between two points of the grid). A number $M = l \times w$ of stations are positioned at regular intervals on this grid and the distance to go from one to another is computed thanks to the Manhattan distance in time. There is a unique station capacity $\mathcal{K} = 10$ and a number $N = M \times V_p \times \mathcal{K}$ of vehicles with V_p being the proportion of vehicles per station.

Demand In BSS data-mining studies, such as Côme (2012), demand appears to be regular along the weeks for a same season. We focus hence on a typical week day that we approximate as schemed in Figure 6.3a: A day lasts 12 hours (say from 6h00 to 18h00). At the end of each day, all vehicles must return to a station. We take as base a fully homogeneous city, *i.e.* the demand is the same for all trips: $\Lambda^t_{a,b} = \Lambda,\ \forall (a,b) \in \mathcal{D},\ \forall t \in \mathcal{T}$. We only consider one way trips: $\Lambda^t_{a,a} = 0,\ \forall a \in \mathcal{M},\ \forall t \in \mathcal{T}$. So, when the proportion of vehicles in the system equals 1 no trip can be sold.

Instance "$M\_w{\times}l\_I\Lambda_s\_[G\Gamma]\_[T\Theta]$" has to be read as follows: it is an homogeneous city with M stations spread on a grid of size w times l, with a demand intensity Λ_s per station per minute ($\Lambda_s = (M-1) \times \Lambda$) and with possibly a gravitational effect of intensity Γ or a tide effect of intensity Θ.

Gravitation pattern We introduce a gravitation phenomenon of factor Γ. It increases by a factor Γ the demand for trips going from stations $\mathcal{L}$ to stations $\mathcal{R}$, decreasing the opposite demand by the same factor Γ, *i.e.* $\Lambda_{a,b} = \Gamma \times \Lambda$ and $\Lambda_{b,a} = \Gamma^{-1} \times \Lambda$ for $(a,b) \in \mathcal{L} \times \mathcal{R}$ and $\Lambda_{a,b} = \Lambda$ otherwise. In the following we use a gravitation of intensity $\Gamma = 3$.

Tides pattern We introduce a morning and an evening tide of intensity Θ. The demand pattern is represented Figure 6.4. The day is divided into three periods, the morning from 6h to 9h, the middle of the day from 9h to 15h and the evening from 15h to 18h. The city is split into two equal sub grids: $l_i \in \mathcal{L}$ and $r_i \in \mathcal{R}$.

1. In the morning there are Θ times more demands than normal for trips going from stations $\mathcal{L}$ to stations $\mathcal{R}$, Θ^2 less in the opposite direction and between stations within $\mathcal{R}$, *i.e.* $\Lambda^{[6,9]}_{l_1,l_2} = \Lambda$, $\Lambda^{[6,9]}_{l_1,r_1} = \Theta\Lambda$ and $\Lambda^{[6,9]}_{r_1,l_1} = \Lambda^{[6,9]}_{r_1,r_2} = \Theta^{-2}\Lambda$.
2. In the middle of the day, there is no demand between $\mathcal{L}$ and $\mathcal{R}$, and Θ^2 less demands between stations within $\mathcal{L}$, *i.e.* $\Lambda^{[9,15]}_{l_1,r_1} = \Lambda^{[9,15]}_{r_1,l_1} = 0$, $\Lambda^{[6,9]}_{l_1,l_2} = \Theta^{-2}\Lambda$ and $\Lambda^{[9,15]}_{r_1,r_2} = \Lambda$.
3. In the evening, there is an opposed tide as in the morning from $\mathcal{R}$ to $\mathcal{L}$, *i.e.* $\Lambda^{[15,18]}_{r_1,r_2} = \Lambda$, $\Lambda^{[15,18]}_{r_1,l_1} = \Theta\Lambda$ and $\Lambda^{[15,18]}_{l_1,r_1} = \Lambda^{[15,18]}_{l_1,l_2} = \Theta^{-2}\Lambda$.

In the following we use a tide of intensity $\Theta = 6$. We study a modification of this tide phenomenon where the evening tide is not the symmetric of the morning tide: $\Lambda^{[15,18]}_{l_1,r_1} = 0$ instead of $\Theta^{-2}\Lambda$. Instances with this modification have *Mod* in their name: for instance we study instance 24_4x6_I0.3_T6_Mod.

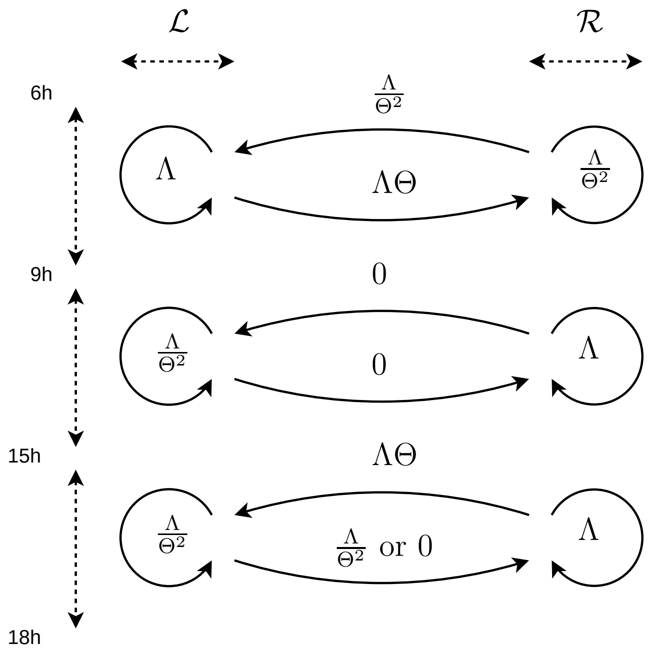


Figure 6.4: Demand pattern for a tide with intensity Θ.

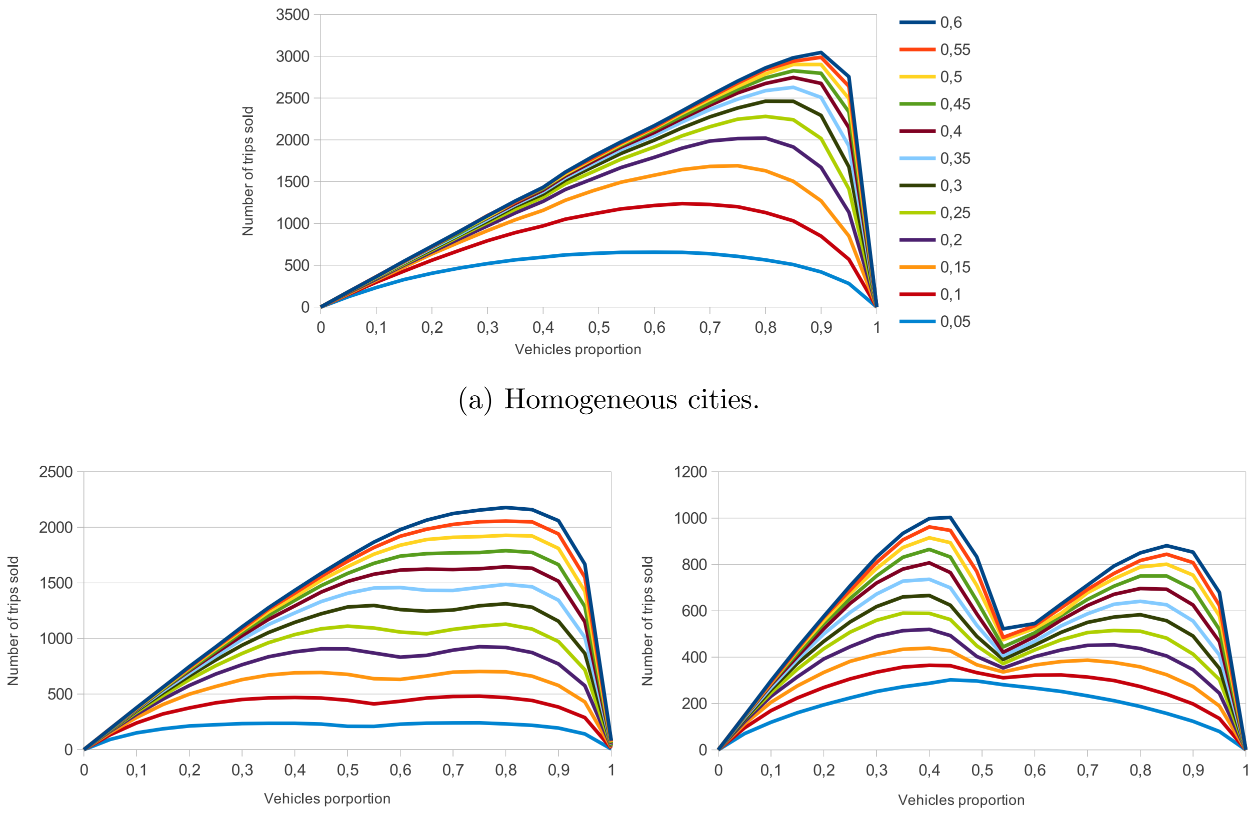
Normalization To decorrelate the tide and the gravitation phenomenon from the simple increase of demands, we normalize the overall demand to keep the same amount of demands as in a full homogeneous city, *i.e.* the expected number of trip requests per day is the same for instances 24_6x4_I0.3, 24_6x4_I0.3_T6 and 24_6x4_I0.3_G3.

6.3.3 Sizing

Demand intensity and fleet sizing To simulate the behaviour of a VSS we need to set the number of vehicles available. Fricker and Gast (2012) study the relationship between demand intensity and the vehicles proportion V_p in function of the station capacity $\mathcal{K}$. For a perfect homogeneous city with an arrival rate Λ_s per station and a unique stochastic transportation time of mean μ^{-1}, the best sizing for a system without any control is $V_p = \frac{1}{\mathcal{K}}(\frac{\mathcal{K}}{2} + \frac{\Lambda_s}{\mu})$. Contrary to them, we consider a protocol with reservation of parking spot at destination and in our homogeneous cities the transportation time is not unique. Nevertheless, in Figure 6.5a we observe a similar dependence to the demand intensity: The more intense the demand is, the higher the vehicles proportion needs to be[4].

When considering unbalanced cities, with gravitation phenomenon, in Figure 6.5b we observe a mustache effect with two local optima. It corroborates the experience

4. Intuitive results without parking spot reservation.



(a) Homogeneous cities.

(b) Cities with gravitation.

(c) Cities with tide.

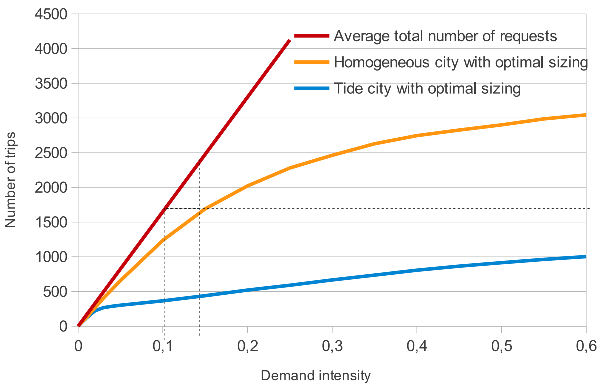
Figure 6.5: Fleet sizing with different demand intensities.

of Fricker *et al.* (2012) with unique transportation times and no reservation protocol. With a tide phenomenon, we also observe a similar mustache in Figure 6.5c. The best vehicle proportion depends on the demand intensity, ranging around 45%.

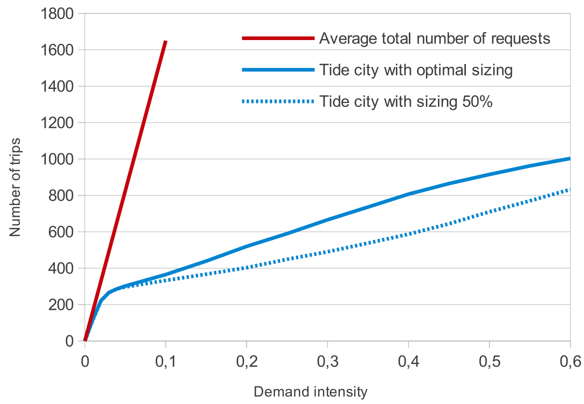
George and Xia (2011) prove that for infinite station capacities the number of trips sold is concave in function of the number of vehicles. When considering station capacities, for non homogeneous cities, in Figure 6.5b and 6.5c we observe that the function does not seem to be concave anymore.

Such variations of the VSS performance indicate that a proper fleet sizing has to be considered when studying other leverages.

A reasonable demand? Figure 6.6a represents the number of trips sold in function of the demand intensity for an homogeneous city and a tide city with their optimal fleet sizing. The number of trips sold is compared to the total average number of requests (0.1 client per station per minute is equal to 1500 requests per day for a system with 24 stations). We observe that for both cities the number of



(a) Number of trips sold for cities with an optimal fleet sizing compared to the average total number of requests.



(b) The number of trips sold in function of the demand intensity is not concave for a given fleet size.

Figure 6.6: Number of trips sold in function of demand intensity: A flat function.

trips sold seems to be concave when considering the best sizing for each intensity. However, in Figure 6.6b we observe that for a given proportion of vehicles, in tide cities, it does not seem to be concave anymore.

In Vélib' there are approximately 150 000 trips sold per day for about 1400 stations. Considering that the majority of these trips are made during 18 hours of the day it gives approximately an arrival intensity of 0.1 clients per station per minute. This number of trips represents the satisfied demand, without any special pricing policy. Figure 6.6a presents simulation results comparing the number of trip requests (demand) and the number of trips sold (satisfied demand). As shows the dotted lines, serving 0.1 clients per minute amounts to serving $\approx$ 1750 clients per day. In an homogeneous city, serving 0.1 clients per minute would hence need an actual demand around 0.15 clients per minute. In a tide city, the function trip sold/demand is almost flat. With such demand pattern it is not even sure that there exists a demand intensity able to serve 0.1 clients per minute.

6.4 Is there any potential gain for pricing policies? An experimental study

6.4.1 Experimental protocol

We first only consider the fluid model, the stable fluid model and the Max Flow With Reservation upper bound. In a second time we will see that Maximum

Circulation static policy and its upper bound are dominated by the stable fluid ones.

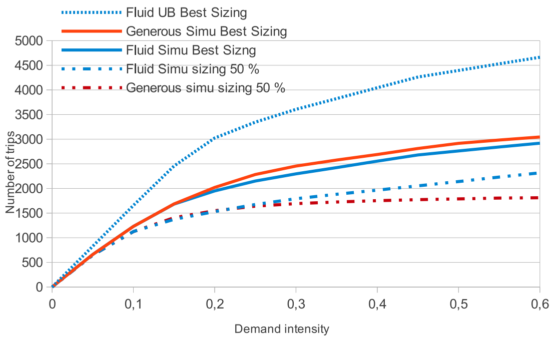
Optimizing the Number of trip sold We focus on optimizing the number of trips sold by the system. We consider a continuous elastic demand with a maximum demand Λ, *i.e.* for each trip, there exists a price to obtain any demands $\lambda \in [0, \Lambda]$. We take as reference the number of trips sold by the *generous* policy, setting on each trip the demand to its maximum value $\lambda = \Lambda$ (all prices to their minimum value). We evaluate the performance in term of number of trips sold of two pricing policies and three Upper Bounds (UB):

1. The fluid SCSCLP (5.2) model (*Fluid*) gives a static policy and an UB conjectured for dynamic policies and time-varying demand (see Chapter 5, page 114).
2. The stable fluid (5.3) pointwise stationary approximation (*S-Fluid*) gives a static policy and an UB on dynamic policies for stable demand (see Chapter 5, page 114).
3. Max Flow With Reservation gives an UB on dynamic policies by optimizing a posteriori the realization of the demands, a *scenario* (see Chapter 3, page 73).

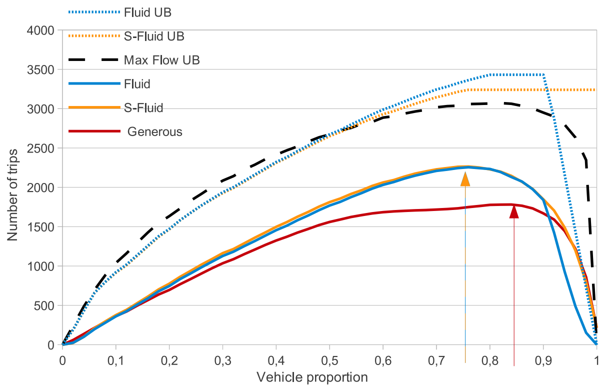
Notice that in practice, the maximum demand Λ might be obtained at a negative price (paying the user), and we should rather optimize the trade off between the number of trips sold and the generated gain but this is beyond the scope of this study.

Simulation We use a real-time station-to-station reservation protocol, *i.e.* users have to book a parking spot at destination before taking a vehicle. We compare our 2 pricing policies and 3 UBs to the generous policy on the same scenario: a simulation of the stochastic evaluation model on 300 days with similar demand patterns. For our instances, the policies tested have only one single strongly connected component, therefore the vehicle are uniformly distributed among the open stations at the beginning of the horizon. Then a 10 days warm up is used as mixing time.

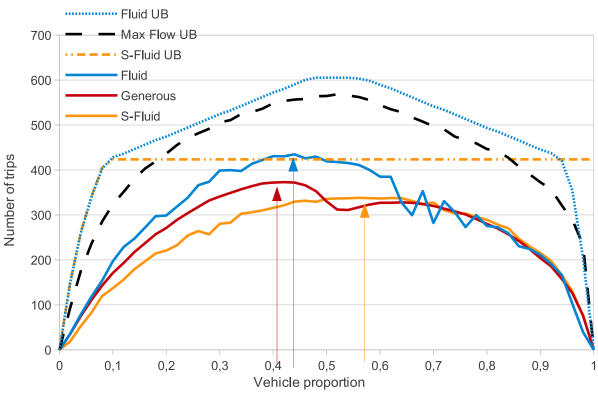
Figure 6.7 reports the number of trips sold by the different pricing policies on instances containing 24 stations of capacity $\mathcal{K} = 10$. The best sizing of each pricing policy is indicated by an arrow. In Figure 6.7a and 6.7b the demand is stationary, therefore Fluid and S-Fluid are almost equivalent. The little difference is due to the off period (night) between two following days considered by Fluid but not by S-Fluid. In Figures 6.7c and 6.7d, we introduce a tide phenomenon implying hence time-varying demands. The value given by stable fluid solution method is hence not giving an upper bound anymore.



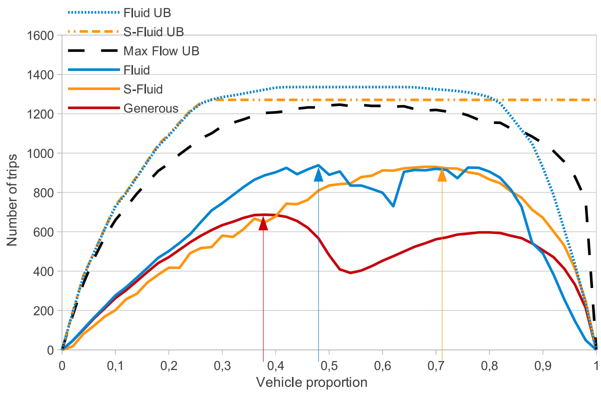
(a) Varying demand intensity: Instances 24_4x6_I0.1-0.6.



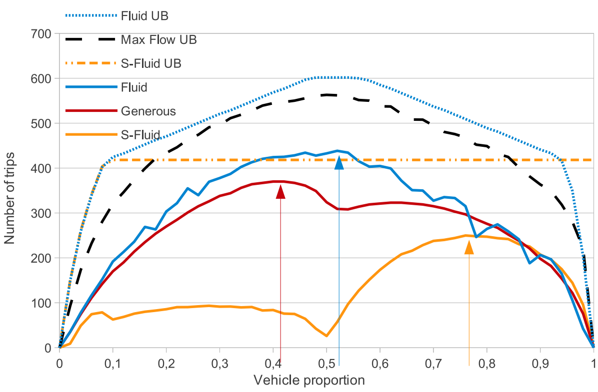
(b) Gravitation: Instance 24_4x6_I0.3_G3.



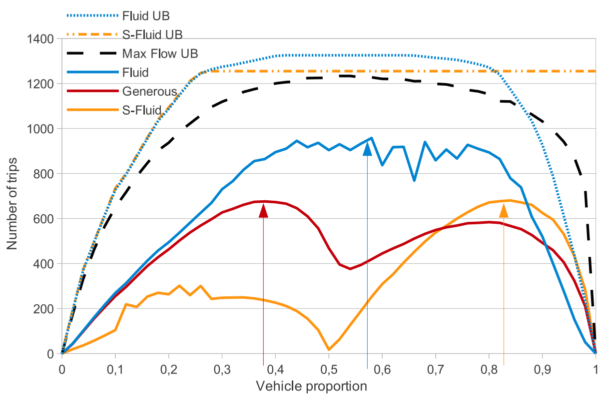
(c) Tide low demand: Instance 24_4x6_I0.1_T6.



(d) Tide higher demand: Instance 24_4x6_I0.3_T6.



(e) Tide low demand: Instance 24_4x6_I10.1_T6_Mod.



(f) Tide higher demand: Instance 24_4x6_I0.3_T6_Mod.

Figure 6.7: Sizing the number of vehicles in the system with a pricing regulation.

6.4.2 Preliminary results

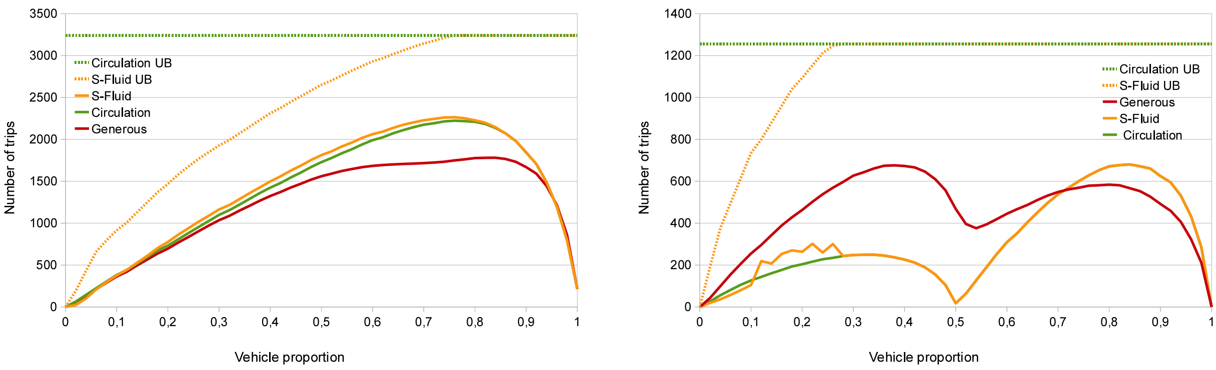
Influence of the demand intensity We look at the influence of the demand intensity in an homogeneous city. In Figure 6.7a we compare the performance of the generous policy and the fluid heuristic policy (Fluid≈S-Fluid) in homogeneous cities with different intensities. Each policy is simulated either with its best fleet sizing computed greedily or with a vehicle proportion of 50%.

With an optimal fleet sizing, the generous policy dominates strictly the fluid policy. But for a given fleet sizing, here filling 50% of the parking spot, the performance of the fluid policy is related to demand intensity: the higher the demand intensity is, the higher the improvement of the fluid heuristic will be. We explain this phenomenon as follows: in an homogeneous city, the only leverage available is to use the difference in transportation times. If the fleet sizing is not optimized for the demand intensity the fluid heuristic increases the number of trips sold by the system by favoring short distance trips.

Influence of the gravitation In Figure 6.7b we compare the performance of the generous policy, the fluid and stable fluid heuristic policies on a city with gravitation. Fluid and S-Fluid are drown on this figure to show that they are almost equivalent. We see that applying fluid policies provides a transit increase of roughly 30% while the UB for any dynamic policy is around 70%.

Influence of the tide In Figure 6.7c and 6.7d we study fluid policies optimization on a tide city with two different intensities. Notice that since we are considering time-varying demand S-Fluid is not giving an UB anymore. For a demand with low intensity (Figure 6.7c), the fluid heuristic increases the number of trips sold by 13% while S-Fluid decreases it. With a higher intensity (Figure 6.7d), the Fluid heuristic increases by 40% the transit of the generous policy. S-Fluid heuristic policy behaves well on this instance selling almost as many trips as Fluid heuristic. However, Fluid attains this best performance with a third less of vehicles.

S-Fluid results instability for time-varying demand With a slight modification in the tide city demand, replacing a very small demand $\Lambda\Theta^{-2}$ by a null one, we obtain totally different results for the S-Fluid heuristic. Figure 6.7e and 6.7f represents the fluid heuristics performance for this modified tide instance. S-Fluid has a really poor transit while generous and Fluid policy behaviours are not that different from the original tide city. It shows that Stable Fluid pointwise stationnary approximation is blind regarding the tide effect and is hence not that stable!



(a) Gravitation: Instance 24_4x6_I0.3_G3.

(b) Tide with high demand: Instance 24_4x6_I0.3_T6_Mod.

Figure 6.8: Stable fluid dominates MAXIMUM CIRCULATION.

Optimization gap We compare the performance of our two upper bounds. MAX FLOW UB seams stronger than Fluid UB. On this benchmark, the difference between the best heuristic policies and the best UB is around 33%. We have tested only static policies but this optimization gap stands also for dynamic policies optimization.

Dominance of stable fluid over MAXIMUM CIRCULATION Like stable fluid model, MAXIMUM CIRCULATION heuristic policy can be used for time-varying demand with a pointwise stationary approximation. Figure 6.8 compares the MAXIMUM CIRCULATION heuristic and Stable Fluid one. We observe that they have almost the same behavior. Stable Fluid policy behaves only slightly better in some cases. Regarding the upper bounds, we remark that when the proportion of vehicles reaches a certain level, 75% for gravitation and 25% with tides, MAXIMUM CIRCULATION UB and stable fluid UB are the same.

6.5 Technical discussions – Models' feature

6.5.1 SCSCLP uniform time discretization

We use a discrete time approximation with time step of fixed length Δ to compute the fluid SCSCLP (5.2) as a linear program. It is a classic way to approximate a CLP. When Δ tends to 0, it is supposed to converge toward the real SCSCLP value. As Δ decreases the objective value (our UB) increases and one can conjecture that the heuristic policy should perform better. We show the contrary.

Experimentally we have tested 4 different time step lengths for the discrete time

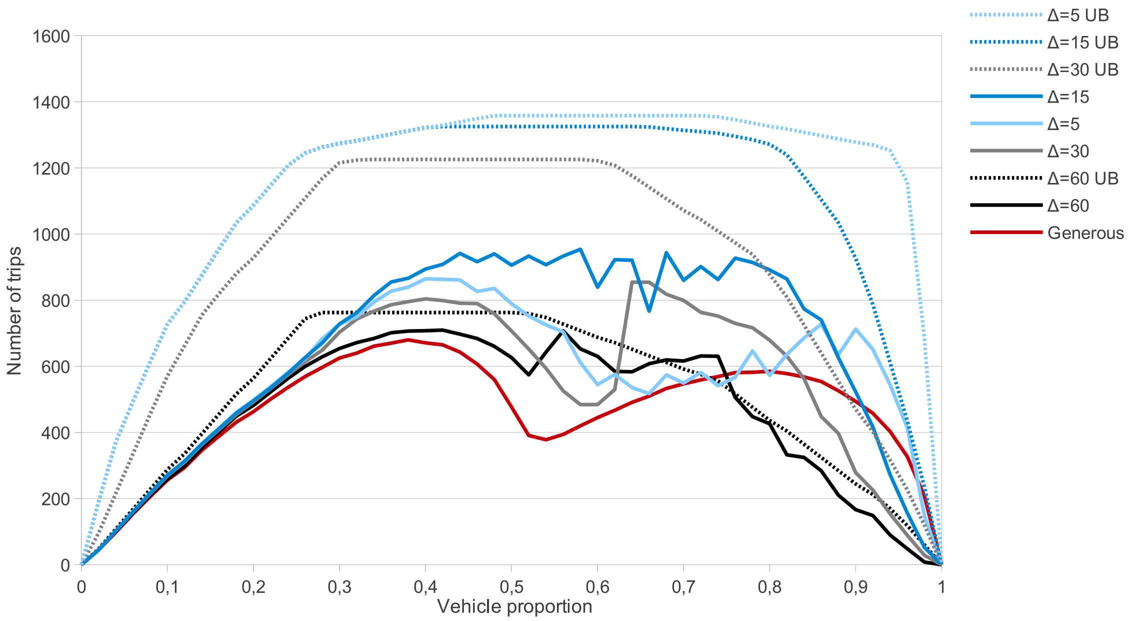
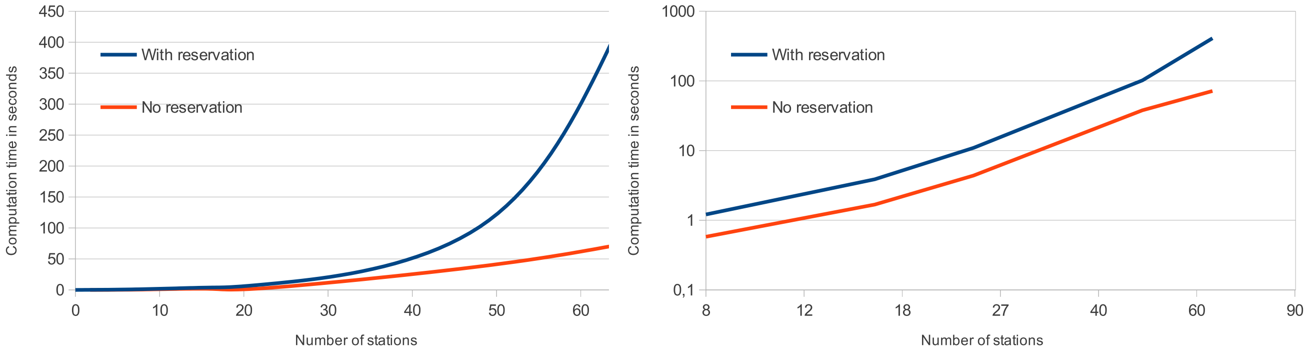


Figure 6.9: Discrete time approximation of fluid SCSCLP (5.2) with different time-step length Δ: Instance 024_4x6_I3_T6.

CLP approximation: $\Delta = 60$, 30, 15 and 5. Figure 6.9 represents the heuristic policy simulation values and the model value (UB) for these four time step lengths on an instance. We make the following observations: When the time step length decreases the UB value increases as it should, to the extent that an approximation with big time step such $\Delta = 60$ UB, is even smaller than the $\Delta = 15$ heuristic policy simulation value. More surprisingly, smaller time steps do not lead to better heuristic policies. Indeed, even if the biggest time step $\Delta = 60$ gives the worst heuristic policy, the smallest one $\Delta = 5$ policy is dominated by $\Delta = 15$ policy, that eventually appears to be the best one. We have two interpretations for having $\Delta = 15$ time step being the best trade off:

1. The fluid model is a deterministic approximation of a stochastic process considering only the average of the demand. When time steps are smaller, the demand rate on a single time step is small and the variance of the stochastic process around the average is then bigger. It is the opposite of the law of large numbers!

2. In our benchmark, transportation times are multiple of 15 minutes, therefore having a time step $\Delta > 15$ implies an overestimation of the transportation times.



(a) Linear scale.

(b) Logarithmic scale.

Figure 6.10: Influence of the reservation constraint on the computation time of the fluid model.

6.5.2 The reservation constraint – Computing time vs quality

Computation time When designing heuristics it is important to consider their abilities to handle real size systems. In Figure 6.10 we compare the computation time of the fluid model with and without the reservation of parking spot constraint. Solving the fluid model with reservation appears much slower in practice (Figure 6.10a), even if it seems to be in the same order of complexity (Figure 6.10b). The same phenomenon is present when comparing the computation time of MAX FLOW WITH RESERVATION and MAX FLOW.

Quality Figure 6.11 compares the performance of the fluid heuristic policy simulation value, the fluid UB and the MAX FLOW UB with and without the reservation constraint. We see that under a vehicle proportion of 30%, considering the parking spot reservation in the model does not produce better heuristics and UBs. Nevertheless, when the percentage of vehicles is over 50%, considering parking spot reservation allows the fluid heuristic to perform much better and the fluid and MAX FLOW UBs to be stronger. It is probably because the parking reservation is less an issue when the proportion of vehicles is low. Notice that when there is one vehicle per parking spot (vehicle proportion=1), only models considering parking reservation predict correctly that 0 trip can be sold.

Conclusion For systems with lots of stations and a vehicle proportion below 30% or 50%, it could be of interest to relax the parking spot reservation constraints

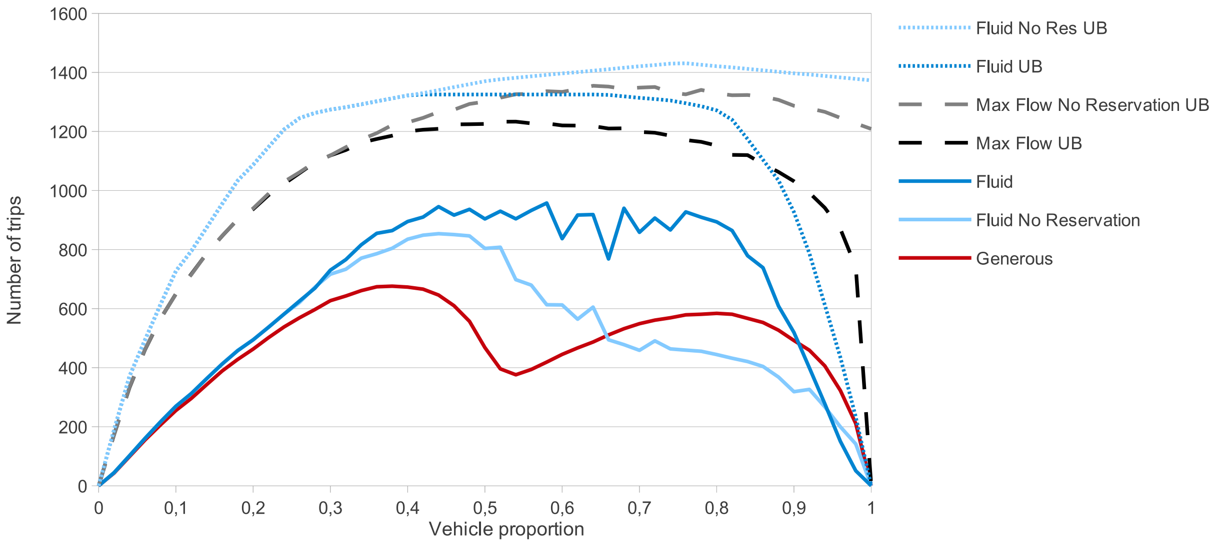


Figure 6.11: Influence of reservation: Instance 024_4x6_I3_T6.

in optimization models in order to gain in computation time keeping a reasonable quality.

6.5.3 Fluid as an ∞-scaled problem

Figure 6.12 tests the s-scaled problem convergence toward the fluid model when s tends to infinity (Conjecture 1 page 118). The generous policy and the fluid heuristic policy are simulated on a s-scaled problem. Their performances are compared to the fluid continuous price model value (Fluid UB) conjectured to be an UB (Conjecture 2 page 119) for all dynamic policies and all scaling s. The number of trips sold by the Fluid UB is constant since the fluid model does not take into account the variance of the demand. We remark that reducing the variance (as s grows) increases the number of trips sold by both policies. The s-scaled problem optimal dynamic policy gain is in between the fluid heuristic policy simulated value and the fluid UB value. The fluid heuristic policy gain seems to converge towards the fluid UB and hence the optimal dynamic value.

For continuous prices optimization, the fluid heuristic policy and the fluid UB are computed thanks to the SCSCLP (5.2). For the generous price policy, we have no efficient algorithm computing the fluid model for one discrete price. However, the generous price policy gain seems also to converge toward a value, that should

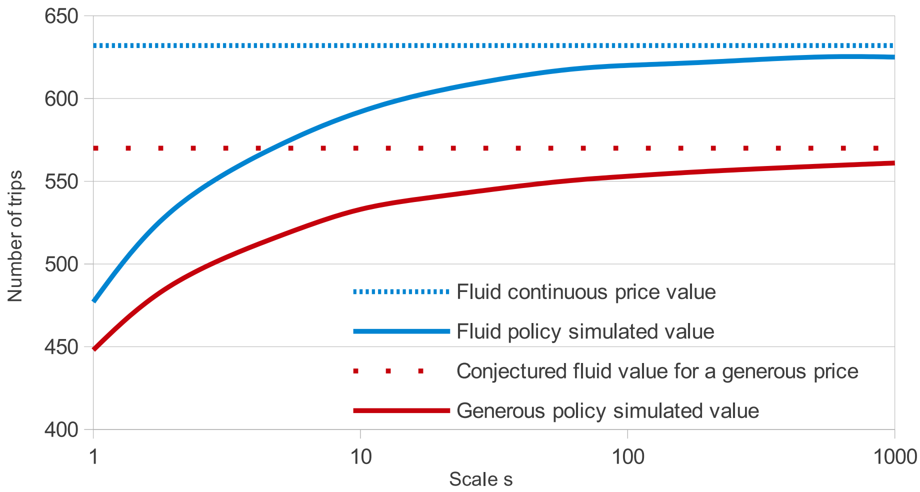


Figure 6.12: Asymptotic convergence of s-scaled problem and fluid model: Instance 4_2x2_I0.3_T3.

be the discrete price (generous price) fluid value.

6.6 Conclusion

We conduced some experimental tests on the pricing heuristic policies and upper bounds proposed in the previous chapters. Our goal was to estimate the potential impact of pricing in VSS. We raised the problem of accessing the real (uncensored) demand that can be used to simulate a city. We showed on a practical case study that using only trips sold historical data leads to considering an "unrealistic" demand, or at least not proper for pricing optimization. Indeed, the fluid upper bound has shown that there were no gap for pricing optimization with such demand. Moreover it is reasonable to think that not 100% of the demand was satisfied in the data; The censored demand is hence not considered and incentive strategies are not applicable.

Since the real demand is not accessible, and to isolate and understand more easily the phenomenons at stake, a simple reproducible benchmark and an experimental protocol was proposed. We exhibited that the pricing leverage needs to be considered jointly with the best fleet sizing. The static fluid heuristic policy appeared to be the best one in the simulations. It allowed to increase between 10% to 30% the number of trips sold. Max Flow With Reservation seemed to provide the stronger upper bound. On the instances tested, optimization gaps for dynamic policies optimization were between 50% to 100%.

We discussed the specificity of the fluid model implementation. We showed a high instability in the fluid approximation's solution method by discrete time approximation. Interestingly, solving the fluid model with 15 minutes time-step discretization provides heuristic policies performing better than those generated when solving it with smaller time-steps. Our explanation is that bigger time-steps are correcting the fluid deterministic approximation by ensuring a minimum demand rate per time-step, reducing hence the (relative) variance of the "estimated" stochastic process.

Conclusion

Learn from yesterday, live for today, hope for tomorrow. The important thing is to not stop questioning.

Albert Einstein (1896–1955)

In English

A research path – Contributions summary

The objective of this thesis is to study the interest of pricing policies for Vehicle Sharing Systems (VSS) optimization. Revenue management and pricing have been studied for other applications in the literature such as airline tickets or internet traffic management. However, the VSS context has specific features: The demand varies quickly along the day but is also pretty regular; The resources are the parking spots as well as the vehicles (with capacity one contrary to airplanes that might have hundreds of seats); The trips sold are interdependent, *e.g.* in order to offer VSS trips from stations a to b you may wish to sell trips going to a at very low price, in order to have available vehicles in a. This is not the case with air tickets where the availability of seats on the flight from a to b is not directly affected by the number of tickets sold from other places to a. To the best of our knowledge, "classic" literature results are hence inapplicable.

VSS management overview In Chapter 1, we gave a general overview of the VSS management. We detailed the specificity of implementing a short term one-way VSS. Current optimization leverages are presented. A formal pricing framework for VSS studies is defined. It has enabled to classify current literature results and to exhibit where our contributions stand.

A stochastic pricing problem In Chapter 2, we proposed a stochastic model to tackle the pricing optimization problem in vehicle sharing systems. This problem is our reference, the "Holy Grail" that we try to solve all along this thesis. This model simplifies reality, though it intends to keep its important characteristics such as time-varying demands, station capacities and the reservation of parking spots at destination. We explained how we can avoid considering explicitly the prices when maximizing the number of trips sold. Indeed, in this thesis, since we focus on maximizing the transit, talking about pricing policies amounts to considering incentive policies or simply policies regulating demand. We proposed a formal definition for the VSS stochastic pricing problem. Although this formulation is compact and relatively simple, solving in general this problem appears hard. Indeed, even mesuring exactly the expected value of a policy seems intractable for real size systems. We discussed notions of complexity in this stochastic framework. A frame is specified in our research of tractable solution methods for the VSS stochastic pricing problem. In this thesis we focus on solution methods with computational complexity polynomial in the number of stations M and the number of vehicles N.

Scenario-based approach In Chapter 3, we investigated a scenario-based approach for the VSS stochastic pricing problem. Its principle is to work a posteriori on a realization of the stochastic process: *a scenario*. Optimizing on a scenario provides heuristics and bounds for the stochastic problem. In this context, such approximation raises deterministic problems with a new constraint: the *First Come First Served constrained flow* (FCFS flow). We presented three such problems: 1) a system design problem, optimizing station capacity and two operational problems setting static prices, 2) on the trips, or 3) on the stations. All three problems were shown APX-hard, *i.e.* inapproximable in polynomial time within a constant ratio. Therefore, we investigated a bound and an approximation algorithm relaxing the FCFS flow constraint based on MAX FLOW WITH RESERVATION. The theoretical guaranty (worst case) is exponential in the number of stations M. Nevertheless, we saw in the simulation that the MAX FLOW WITH RESERVATION upper bound seems competitive in practice. It is even the best upper bound for the dynamic policies optimization available.

Optimizing with product forms In Chapter 4, we restricted our study to a simpler stochastic model. In order to provide exact formulas and analytical insights: transportation times are assumed to be null, stations have infinite capacities and the demand is Markovian stationary over time. This simplified model is still intractable for an explicit dynamic pricing optimization because the number of states to consider is exponential in M and N. We proposed a heuristic based on computing a Maximum Circulation on the demand graph together with a convex integer program solved optimally by a greedy algorithm. For M stations and N vehicles, the performance ratio of this heuristic is proved to be exactly $N/(N+M-1)$. Hence, whenever the number of vehicles is large compared to the number of stations, the performance of this approximation is very good. For instance for 10 vehicles per station it is leading to an 9/11-approximation.

Several extensions are natural for this work. We believe that adding transportation times has a minor impact on our results. Moreover, since circulation policies spread vehicles very well among the stations, adding capacities to the stations may still allow these policies to be efficient.

Fluid approximation In Chapter 5 we presented a fluid approximation constructed by replacing stochastic demands with a continuous deterministic flow (keeping the demand rate). The fluid dynamic is deterministic and evolves as a continuous process. The fluid model has for advantage to consider time-varying demand. We showed that solving it with discrete prices seems difficult (inducing non-linearity). For continuous prices, we proposed a fluid approximation SCSCLP formulation maximizing the transit. The solution of this program produces a static policy. The optimal value of this SCSCLP is conjectured to be an upper bound on dynamic policies. For stationary demand the fluid model is formulated as a linear program. It produces a static heuristic policy and the value of this LP is proved to be an upper bound on dynamic policies optimization. The stationary fluid model can be used for time-varying demand with a piecewise stationary approximation.

Simulation In Chapter 6 we tried to estimate the potential impact of pricing in VSS. We tested the heuristic policies presented in the previous chapters on case studies. A practical case study was conduced on Capital Bikeshare historical data. A simple demand pattern was generated from these data. We showed that for such demand there is no potential gain for pricing policies. It exhibits the problem of accessing the real demand. We proposed a simple reproducible benchmark and an experimental protocol. We exhibited that the pricing leverage needs to be considered jointly with the best fleet sizing. The static fluid heuristic policy appeared to be the best one on the simulations. It allowed to increase from 10% to 30% the number of trips sold. Max Flow With Reservation provided the best upper bound. Optimization gaps for dynamic policies optimization we from 50% to 100%.

Perspectives

Fluid model modification The fluid heuristic policy is the one providing the best performance in our simulations. However this heuristic suffers from instability with the discrete-time solution method (see Section 6.5.1, page 136). Interestingly, solving the fluid model with 15 minutes time-step discretization provides heuristic policies performing better (in our simulations) than those generated when solving it with smaller time-steps. Our explanation is that bigger time-steps are correcting the fluid deterministic approximation by ensuring a minimum demand rate per time-step, reducing hence the (relative) variance of the "estimated" stochastic process. Nevertheless solving it optimally is the only way to provide the (conjectured) "real" upper bound on optimization.

To strike the deterministic approximation optimism, one should maybe penalize problematic states where the stations are expected to be nearly empty (resp. full) by reducing the demand intensity of the outgoing (resp. ongoing) demand. To do so, one can assume the independence of each station and hence consider the availability $A_{a,b}$ of a trip (a, b) to be equal to the product of the availability A_a^+ of a vehicle in station a and the availability A_b^- of a parking spot in station b: $A_{a,b} = A_a^+ \times A_b^-$. We could then assume that a station filling follows a truncated geometric distribution. It is not the case in practice but seems to be a descent approximation. With such assumption the fluid model will not be an upper bound anymore but it might improve the fluid heuristic performance. Regarding the solution method, a linear approximation could provide an efficient technique.

Optimizing by simulation dynamic policies with compact forms In our simulations, the fluid model and the Max Flow With Reservation upper bounds have exhibited an important optimization gap for dynamic policies optimization. The static policies proposed in this thesis are unable to cope with this gap. Can dynamic policies close this gap?

An exact tractable optimization of dynamic policies needs a compact formulation. However simple dominant structures seem hard to determine (see Section 2.3.3.2, page 52). A direction of research might be then to investigate simple threshold heuristic policies. Even if they can be suboptimal, they might be efficient in practice. Simulation-based optimization, as in Osorio and Bierlaire (2010), is a heuristic way of optimizing dynamic policies. For such research, the simulation time is the bottleneck for estimating the different policy parameters. We should then restrict to policies with a little number of parameters (variables to set), such as virtual station capacity policies (see Section 2.3.3.2, page 52). Moreover, to obtain quick and efficient results, a convergence study of the stochastic process estimation by simulation should be conduced. Indeed, one might think to an experimental protocol adapting the simulation horizon length in order to: 1) derive roughly in which area searching the parameters, 2) increase this horizon length for better precision.

Considering users' flexibility In this study we focused on a real-time station-to-station protocol that is restrictive and probably unrealistic. In a real-life context, especially with a good information system, users might delay their trips, change origin/destination stations or wait a couple of minutes at a station to take/return a vehicle. A promising direction of research is to study if an optimized management of these spatial and temporal flexibilities can increase the VSS utilization. Two axes of research might be of interest:

1) Decentralized controls where each user acts independently looking for his own interest. Such model needs the definition of an individual user behavior. For instance with a utility functions considering costs for the total travel time, the walking distance... An example a dynamic heuristic policy using such utility function is proposed in Chemla *et al.* (2013).

2) Centralized control studies where the system is directing each user. For instance Fricker and Gast (2012) study a policy where users are giving two destination stations and the system is directing them to the least loaded one. Such controls can be seen as dynamic policies. However, one can doubt that they are realistic in practice: users might be able to cheat to obtain the station they want. Nevertheless, centralized control policies might be easier to optimize and their optimization gap is an upper bound on decentralized one [5].

In this thesis we saw that even without considering any flexibility, an exact optimization of a stochastic VSS model seems already hard. Hence, solving exactly models with flexibility is probably too optimistic. Two directions of research might be investigated then: 1) Checking by simulation the performance of intuitive heuristic policies such as load balancing policies. For instance the power of two choices is studied analytically in Fricker and Gast (2012) and by simulation in Fricker *et al.* (2012). 2) Solving exactly simple game theory models trying to capture the important features. The idea is then to derive heuristic policies that are tested by simulation on more realistic models.

Implementing policies in practice In this thesis we investigated whether pricing policies can improve vehicle sharing systems utilization. One can wonder how

5. The difference between the best centralized and the best decentralized policy can be seen as the price of anarchy.

applicable in real-life is a dynamic policy or a static policy changing every hours. Continuous prices are convenient to optimize but might have an important cognitive cost for the users. These complex optimization mechanisms might finally deter them from using the system. However, there exist simple ways to implement such policies in practice. For instance, for transit optimization, a continuous pricing policy is just an aimed demand $\lambda \in [0, \Lambda]$. The system can reach this demand by setting the prices to their minimum values ($\lambda = \Lambda$) and then implement a *probabilistic coin-flip policy*[6], *i.e.* to obtain a demand $\lambda = \Lambda/X$, the system accepts randomly one client out of X. Or if the price $p(\Lambda)$ to obtain demand Λ is negative, which means that the system actually needs to pay the user to take a trip, the system could set let say three discrete prices to propose according to a dynamic (probabilistic) policy: *e.g.* $p(\lambda/2)$, $p(\lambda)$ and $p(\Lambda)$. Moreover, a fundamental assumption of our study is the reservation of parking spot at destination. For such protocol, even if users can see the current number of vehicles and free parking spots at any station (through a communication system), they are blind regarding possible existing reservations. Therefore, if the system tells them that the trip they wish to take is unavailable, they will not have any other choice than to believe it!

A global project In our simulations we raised the problem of having a proper benchmark to estimate the interest of pricing policies. How can we become more credible and give more accurate answers to decision makers? For more convincing results such study has to be part of a broader project involving researchers from different domains. To direct the research toward the most realistic and useful direction, they would have to work together going back and forth between models adapting them. For such global project, one might think of the following task/module decomposition:

- **A)** *System modeling and simulation.* Micro-description of one-way VSS dynamics. Generalization of the utilization contexts including car sharing systems, bike sharing systems, car/truck rentals... Development of a generic simulator integrating the different leverages of optimization. Proposition of performance indicators. Chapter 1 is somewhat a preliminary study for such task.
- **B)** *Formalizing and collecting data.* Creation of a generic format to store VSS historical data. Explicit the importance of each information. Raise awareness of VSS operators on the necessity of giving good data. Collect those data.
- **C)** *Demand (re)building.* Estimate the real-demand for VSS. This demand can

6. Systems might also need to identify users individually in order to avoid having the same one asking for a trip recursively.

be built with historical data (Rudloff *et al.*, 2013) but also by crossing other information. Definition of user behavior models including spatial, temporal and price flexibilities.

D) *Demand analyses and dimension reduction.* Isolation of the core of VSS demand. Characterization of phenomenons. An example is the station/trip clustering done in data-mining literature (Côme, 2012). Development of toy/simple (open source) benchmarks.

E) *Mathematical optimization.* Using operation research, develop tractable solution methods to improve VSS performance. Characterize the range of action of the different leverages. Propose to decision makers decision support systems based on demand generated by modules C) or D).

F) *Real-life experimental studies* Partnership with system operators. Confront models to real-life experiences. Go back and forth on assumptions and results of modules A) to E).

As a conclusion, in this thesis we have mainly worked on modules A) System modeling and simulation and E) Mathematical optimization.

(Conclusion) En français

Une histoire de recherche – Résumé des contributions

Cette thèse a pour objet d'étudier l'intérêt des politiques tarifaires pour optimiser les systèmes de véhicules en libre service en aller-simple, *Vehicle Sharing Systems* (VSS) en anglais. Dans la littérature, les techniques de revenue managements et l'application de politiques tarifaires ont été étudiées pour d'autre contextes tel que les ventes de billets d'avion ou la gestion du trafic internet. Cependant, le cas des VSS a ses spécificités propres. Les demandes varient rapidement au cours de la journée ; Les ressources sont désormais autant les places de parking que les véhicules (avec une seule place contrairement aux avions qui peuvent transporter une centaine de passagers) ; Les trajets vendus sont interdépendants, *e.g.* pour pouvoir offrir des trajets entre les stations a et b on a peut-être intérêt à vendre des trajets vers la station a à des prix très faibles, de manière à avoir des véhicules disponibles en a. Ce n'est pas le cas pour les billets d'avion où la disponibilité des sièges sur un vol de a à b ne dépend pas directement du nombre de tickets vendus depuis d'autre endroits vers a. À notre connaissance, les résultats "classiques" de la littératures sont donc inapplicables.

Gestion des VSS Le Chapitre 1 a présenté un aperçu général sur la gestion des systèmes de véhicules en libre service. Nous avons discuté des spécificités d'implémentation des VSS avec location courte durée en aller simple. Les leviers d'optimisations actuels ont été présentés. Un cadre formel pour l'optimisation de politiques tarifaires a été permis de présenter une revue de littérature classifiée, permettant de situer nos contributions.

Un problème stochastique de tarification Le Chapitre 2 a présenté un problème stochastique de tarification dans les systèmes de véhicules en libre service. Ce problème est notre référence. Sa résolution est le "Graal" que nous poursuivons tout au long de cette thèse. Il simplifie la réalité tout en essayant de conserver ses caractéristiques importantes telles que les demandes variant au cours du temps, les capacités des stations et la réservation d'une place de parking à destination. Nous avons expliqué comment il est possible d'éviter de considérer de manière explicite les prix pour certains objectifs comme la maximisation du nombre de trajets vendus. Puisque le nombre de trajets vendus est le critère retenu pour notre étude, nous pouvons finalement parler autant de politiques incitatives, de régulation de la demande que de politiques tarifaires. Nous avons proposé une définition formelle du problème stochastique de tarification. Bien que cette formulation soit compacte et relativement simple, résoudre ce problème de manière général parait difficile. En effet même mesurer exactement la valeur d'une politique semble *intractable* pour des systèmes de tailles réelles. Des notions de complexité dans cet environnement stochastique ont été discutées. Un cadre de recherche a été spécifié : nous cherchons des méthodes avec une résolution de complexité polynomiale en fonction du nombre de stations M et de véhicules N.

Approche par scénario Dans le Chapitre 3, nous avons étudié une approche par scénario, *i.e.* une optimisation déterministe hors ligne sur une réalisation d'un processus stochastique (*un scénario*). Ce modèle déterministe peut être utilisé pour fournir des heuristiques et des bornes sur le problème d'optimisation en ligne. Cette approche a soulevé une nouvelle contrainte le *flot premier arrivé premier servi*. Nous avons présenté trois problèmes basés sur cette contrainte : un problème stratégique, l'optimisation de la taille des stations, et deux problèmes opérationnels calculant des politiques tarifaires statiques. Nous avons montré qu'ils sont tous trois APX-hard, *i.e.* inapproximable en temps polynomial en dessous d'une certaine constante. Nous avons étudié une borne supérieure sur toutes les politiques dynamiques basée sur le calcul d'un Flot Max. Sa performance a été prouvé faible dans le pire cas. Cependant, dans nos simulations, cette borne supérieure est apparu la meilleure dont nous disposons. Nous avons prouvé que le Flot Max peut également donner un algorithme d'approximation de faible performance (théorique et pratique) mais intéressant pour caractériser la complexité du problème d'optimisation.

Optimisation avec des formes compactes Dans le Chapitre 4, nous nous sommes restreint à l'étude d'un modèle stochastique simplifié. De manière à obtenir des formules exactes et des résultats analytiques, les temps de transports sont considérés instantanés, les stations ont des capacités infinies et la demande est markovienne stationnaire. Ce modèle est toujours *intractable* pour une optimisation explicite car le nombre d'états à considérer est exponentiel en M et N. Nous avons donc proposé une politique heuristique basée sur le calcul d'une Circulation Maximum sur le graphe des demandes couplé à un programme entier convexe résolu optimalement par un algorithme glouton. Pour M stations et N véhicules, le ratio de performance de cette heuristique est prouvé être exactement $N/(N + M - 1)$. Par conséquent, lorsque le nombre de véhicules est grand devant le nombre de stations, la performance de cette approximation est très bonne.

Plusieurs extensions sont naturelles pour ce travail. Nous pensons qu'ajouter des temps de transport a un impact mineur sur nos résultats. De plus, puisque les politiques de circulation repartissent bien les véhicules entre les stations, ces politiques peuvent être efficace même en considérant des capacités de stations.

Approximation fluide Dans le Chapitre 5 nous avons présenté une approximation fluide (déterministe) du processus markovien que l'on peut voir comme un problème de plomberie. Le modèle fluide est construit en remplaçant les demandes discrètes stochastiques par des demandes continues déterministes égales aux valeurs des espérances. Les véhicules sont considérés comme un fluide continu, dont la répartition entre les stations évolue de manière déterministe dans un réseau de réservoirs inter-connectés par des tuyaux. Nous avons montré que résoudre le modèle fluide avec des prix discrets induit de la non-linéarité. Pour des prix continus, nous avons montré qu'optimiser le débit de ce système peut se formuler comme un programme linéaire continu, de type *State Constrained Separated Continuous Linear Program* (SCSCLP), qui peut se résoudre de manière efficace. La solution de ce programme fournit une politique statique. La valeur optimale de ce SCSCLP est conjecturée être une borne supérieure sur toutes les politiques dynamiques.

Simulation Dans le Chapitre 6 nous avons essayé d'estimer l'impact potentiel des politiques tarifaires dans les systèmes de véhicules en libre service. Nous avons donc testé sur des cas d'études les politiques heuristiques ainsi que des bornes supérieures proposées dans les chapitres précédents. Un cas d'étude réel a été analysé sur les données d'exploitation de Capital Bikeshare. Un patron de demande simple a été extrapolé. Nous avons montré que pour une telle demande il n'y avait pas de gain d'optimisation. Cela a mit en exergue la nécessité d'accéder à la demande réelle. Un *benchmark* simple et reproductible ainsi qu'un protocole expérimental a été proposés. Nous avons montré que l'étude des politiques tarifaires doit se faire conjointement avec un dimensionnement optimal de la flotte de véhicules. La politique statique donnée par l'approximation fluide a apparu être la meilleure dans nos simulations. Elle a permis de d'améliorer de 10% à 30% le nombre de trajets vendus. Le borne supérieur basée sur le Flot Max est apparue être la plus forte. Des gains potentiels d'optimisations de l'ordre de 50% à 100% ont été observés pour les politiques dynamiques.

Perspectives

Modification du modèle fluide La politique heuristique fournie par le modèle fluide est celle qui a procuré les meilleurs résultats dans nos simulations. Cependant cette heuristique souffre d'instabilité lorsque l'on résout le modèle continu avec une approximation à temps discret (voir Section 6.5.1, page 136). Il est intéressant de noter que résoudre le modèle fluide avec une discrétisation en pas de temps de 15 minutes produit des politiques heuristiques plus performantes (dans nos simulations) que celles produites lorsqu'on le résout avec une plus petite discrétisation. Notre explication est que de "gros" pas de temps corrigent l'approximation déterministe en s'assurant un taux minimum de demande par pas de temps, réduisant ainsi la variance (relative) du processus stochastique estimé. À noter cependant que résoudre optimalement le modèle fluide est la seule façon de calculer une borne supérieure (conjecture) sur toutes les politiques dynamiques.

Pour palier à l'optimisme de l'approximation déterministe, peut-être devrait-on pénaliser les états problématiques où les stations sont prévues être presque vides (resp. pleines) en réduisant l'intensité de la demande de départ (resp. d'arrivée). Pour ce faire nous pouvons supposer l'indépendance de chaque station et considérer que la disponibilité $A_{a,b}$ d'un trajet (a, b) est égale au produit de la disponibilité A_a^+ d'un véhicule à la station a et la disponibilité A_b^- d'une place de parking à la station b : $A_{a,b} = A_a^+ \times A_b^-$. Nous pourrions ainsi supposer que le remplissage d'une station suit une loi géométrique tronquée. Ce n'est pas le cas en pratique mais cela parait une bonne approximation. Avec de telles hypothèses, le modèle fluide ne serait plus une borne supérieure mais la politique heuristique fluide serait peut être améliorée. En ce qui concerne la méthode de résolution, une approximation linéaire par morceaux pourrait s'avérer efficace.

Optimiser par simulation des politiques dynamiques compactes Dans nos simulations les bornes supérieures fournies par le fluide et le Flot Max ont montré un important potentiel d'optimisation pour les politiques dynamiques. Les politiques statiques proposées dans cette thèse ont été incapable de réduire cet écart. Est-ce qu'une politique dynamique, même simple, pourrait obtenir de meilleures performances ?

Optimiser de manière exacte et efficace (sans expliciter tous les états du système) les politiques dynamiques nécessiterait de caractériser leurs structures pour pouvoir les modéliser sous une forme compacte. Malheureusement nous n'avons pas été capable de faire ressortir de telles structures (voir Section 2.3.3.2, page 52). Une perspective de recherche pourrait être d'optimiser des politiques par seuil "simple". En effet, même si elles sont en général sous-optimales, en pratique elles pourraient donner de bons résultats. L'optimisation basée sur la simulation, à l'instar de Osorio and Bierlaire (2010), est une façon heuristique d'optimiser des politiques dynamiques. Pour une telle optimisation, le temps nécessaire à la simulation est le nerf de la guerre dans l'estimation des paramètre des politiques. Il faudrait surement se limiter à des politiques avec peu de paramètres comme par exemple les politiques définissant des capacités virtuelles (voir Section 2.3.3.2, page 52). De plus, pour obtenir rapidement et efficacement des résultats, une étude de la convergence de l'estimation du processus stochastique par simulation devrait être conduite. En effet, ce serait nécessaire pour établir un protocole expérimental adaptant dynamiquement l'horizon de la simulation afin de maitriser la vitesse de convergence vers un bonne solution : 1) dégrossir dans quel champs de valeurs chercher les paramètres, 2) agrandir l'horizon de simulation pour obtenir une plus grande précision.

Considérer la flexibilité des utilisateurs Dans cette étude nous nous sommes focalisé sur un protocole de réservation en temps réel pour des trajets entre deux stations. Ceci est restrictif et probablement non réaliste. Dans un contexte réel, spécialement avec les moyens de communications actuels, les utilisateurs peuvent retarder leur trajet, changer leurs stations d'origine/de destination ou encore attendre quelques minutes à une station pour prendre/retourner un véhicule. Une direction prometteuse de recherche est la considération de ces flexibilités spatiales et temporelles. Deux axes de recherches se dégagent alors :

1) Les contrôles décentralisés où les utilisateurs agissent indépendamment cherchant chacun leur propre intérêt. Formaliser ces contrôles nécessite la définition du comportement individuel des utilisateurs. Par exemple en utilisant une fonction d'utilité considérant des couts de transport, de marche à pied, d'attente... Un exemple de politique dynamique heuristique utilisant une fonction d'utilité est proposé par Chemla *et al.* (2013).

2) Les contrôles centralisés où le système dirige lui même chaque utilisateur. Par exemple Fricker and Gast (2012) ont étudié une politique où l'utilisateur donne deux destinations et où le système le dirige vers la station la moins chargée des deux. De tels contrôles peuvent être vus comme des politiques dynamiques. Cependant, on peut douter de leurs pertinences pour un contexte réel (les utilisateurs pourraient tricher pour obtenir la station de leur choix). Néanmoins, les politiques centralisées sont plus faciles à optimiser que les décentralisées, donnant de plus une borne supérieure sur l'optimisation de celles-ci [7].

Dans cette thèse nous avons vu que même en ne considérant aucune flexibilité, une optimisation exacte du modèle stochastique "général" parait dure à résoudre. Par conséquent, vouloir résoudre de manière optimale des modèles considérant de la flexibilité est peut être un peu trop optimiste. Deux directions de recherche

7. La différence entre la meilleure politique centralisée et la meilleure politique décentralisée peut être vu comme le prix de l'anarchie.

sont alors envisageable : 1) Vérifier par simulation la performance de politiques heuristiques intuitives tel que l'équilibre des charges. Par exemple, *the power of two choices* est étudié analytiquement par Fricker and Gast (2012) et par simulation par Fricker *et al.* (2012). 2) Résoudre optimalement des modèles simples de théorie des jeux en essayant de capturer des caractéristiques importantes du problème réel. L'idée est ensuite d'en dériver de politiques heuristiques qui seront testées par simulation sur des modèles plus complexes.

Mise en place de politiques complexes en contexte réel Dans cette thèse nous avons étudié si les politiques tarifaires pouvaient améliorer la performance des systèmes de véhicules en libre service. On est en droit de se demander si une politique dynamique, ou une politique statique avec des prix continus changeant chaque heure, peut être appliquée dans un contexte réel. Les prix continus sont commodes à optimiser mais peuvent avoir un cout cognitif important pour l'utilisateur. De complexes mécanismes d'optimisations peuvent finalement dissuader les utilisateurs d'utiliser le système. Néanmoins il existe des façons simples d'implémenter de telles politiques en pratique. Par exemple, lorsque l'on maximise le transit, une politique acceptant des prix continus revient simplement à fixer un objectif de demande $\lambda \in [0, \Lambda]$. Le système peut atteindre cette demande en fixant les prix au minimums ($\lambda = \Lambda$) et en appliquant une politique *probabiliste*[8], *i.e.* pour obtenir une demande $\lambda = \Lambda/X$, le système accepte alors aléatoirement un client sur X. Ou bien si le prix $p(\Lambda)$ pour obtenir une demande Λ est négatif, c'est à dire que le système doit payer un utilisateur pour effectuer un trajet, le système peut alors définir disons 3 prix discrets à proposer de manière dynamique (probabiliste) : *e.g.* $p(\lambda/2)$, $p(\lambda)$ et $p(\Lambda)$.

Par ailleurs, une hypothèse fondamentale de notre étude est la réservation d'une place de parking à destination. Pour un tel protocole, même si les utilisateurs peuvent voir le nombre de véhicules et de places libres sur n'importe quelle station (grâce à leur smart phone par exemple), ils n'ont pas connaissance des réservations existantes. Par conséquent, si le système dit à un utilisateur que le trajet qu'il désire effectuer n'est pas disponible, il n'a pas d'autres choix que de le croire !

Un projet global Dans nos simulations, nous avons soulevé la difficulté d'établir un *benchmark* pertinent pour estimer l'intérêt potentiel des politiques tarifaires. Comment pourrions nous être plus crédible et donner des réponses plus précises aux décideurs ? Nous pensons que pour des résultats plus convaincant notre étude doit faire partie d'un projet plus large, impliquant des chercheurs de différents domaines.

8. Les systèmes ont peut être également intêret à identifier les utilisateurs individuellement pour ne pas accepter qu'ils demandent le même trajet plusieurs fois de suite.

Pour diriger la recherche vers une direction plus réaliste et utile, cette équipe pluridisciplinaire devrait travailler ensemble pour adapter les modèles et comprendre l'enjeu global. Pour un tel projet, la décomposition en tâches/modules suivante pourrait être envisagée :

A) *Modélisation et simulation du système.* Micro-description du fonctionnement d'un système de véhicules en libre service en aller simple. Généralisation du contexte d'utilisation incluant les locations de voitures, vélos, camions... Développement d'un simulateur générique intégrant les différents leviers d'optimisations. Proposition d'indices de performance. Le Chapitre 1 est d'un certaine façon une étude préliminaire de ce module.

B) *Recensement, formalisation et collecte des données d'exploitation utiles.* Création d'un format générique pour stocker les données d'exploitation potentiellement utiles, en prenant en compte que les données accessibles en générale seront partielles et dépendantes de chaque système réel à l'étude. Collecte de données d'exploitation. Mise au format des données de terrain collectées. Expliciter l'importance de chaque information pour sensibiliser les opérateurs à fournir des données de qualité.

C) *Modélisation de la demande en contexte tarifaire donné et estimations numériques.* Estimer la demande réelle pour un système de véhicules en libre service. Cette demande peut être construit à partir de données historiques (Rudloff *et al.*, 2013) mais aussi en croisant différentes sources d'information. Définition d'un modèle utilisateur incluant ses flexibilités spatiales temporelles et tarifaires.

D) *Analyse de la demande et réduction de la dimension.* Approximation des courbes temporelles en données compactes. Caractérisation des phénomènes de déséquilibre. Un exemple est le *clustering* par station/trajet effectué en *data mining* (Côme, 2012). Développement de *benchmarks* simples (open source).

E) *Optimisation mathématique.* Utiliser les techniques de recherche opérationnelle, développer des méthodes de résolution efficace pour améliorer l'efficacité des systèmes de véhicules en libre service. Caractériser les potentiels gains, rayon d'action, de chacun des leviers d'optimisation. Proposer aux décideurs des systèmes d'aide à la décision basés sur une demande générée par les modules C) ou D).

F) *Études expérimentales en contextes réels* Partenariat avec des opérateurs de systèmes. Confronter les modèles et résultats théoriques à la réalité par expérimentations. Affiner les hypothèses et objectifs des modules A) à E).

Pour conclure, dans cette thèse nous avons principalement travaillé sur les modules A) Modélisation et simulation du système et E) Optimisation mathématique.

Appendices

Appendix A

Action Decomposable Markov Decision Process

One should always generalize.

Carl Gustav Jacobi (1804–1851)

This appendix presents theoretical results to tackle Markov Decision Processes (MDP) with (large) Decomposable action space (D-MDP). Before being generalized, this study was originally motivated by our investigations on the VSS stochastic problem, especially under its simplified form presented in Chapter 4 (null transportation times, infinite station capacities and a stationary demand). Problems raised when we modeled the VSS dynamic discrete pricing stochastic problem as a MDP. The classic MDP model considers, in each state $s \in \mathcal{S}$, a set $\mathcal{Q}$ of discrete prices for each possible trip. MDPs are known to be polynomially solvable in the number of states $|\mathcal{S}|$ and actions $|\mathcal{A}|$ available in each state. However, in each state $s \in \mathcal{S}$, the VSS MDP model's action space $\mathcal{A}(s)$ is the Cartesian product of the available prices for each trip, *i.e.* $\mathcal{A}(s) = \mathcal{Q}^{|\mathcal{M}|^2}$. Hence, the action space size is exponential in the number of stations. To avoid suffering from this explosion, we present in this appendix the action Decomposable Markov Decision Processes (D-MDP): a general framework based on the event-based dynamic programming (Koole, 1998). Modeled as a D-MDP, the complexity of solving the VSS stochastic pricing problem becomes polynomial in $|\mathcal{S}|$ and $|\mathcal{Q}||\mathcal{M}|^2$ (that is far less than $|\mathcal{Q}|^{|\mathcal{M}|^2}$). Nevertheless, another problem is the explosion of the state space $\mathcal{S}$ with the number of vehicles and stations. This phenomenon is known as the *curse of dimensionality* (Bellman, 1953). VSS D-MDP model is therefore unable to solve real-scale instances, but it has still helped us to figure out the complex structure of dynamic optimal policies (see Section 2.3.3.2, page 52).

Chapter abstract

We consider a special class of continuous-time Markov decision processes (CTMDP) that are action decomposable. An action-Decomposed CTMDP (D-CTMPD) typically models queueing control problems with several types of events. A sub-action and cost is associated to each type of event. The action space is then the Cartesian product of sub-action spaces. We first propose a new and natural Quadratic Programming (QP) formulation for CTMDPs and relate it to more classic Dynamic Programming (DP) and Linear Programming (LP) formulations. Then we focus on D-CTMDPs and introduce the class of decomposed randomized policies that will be shown to be dominant in the class of deterministic policies by a polyhedral argument. With this new class of policies, we are able formulate decomposed QP and LP with a number of variables linear in the number of types of events whereas in its classic version the number of variables grows exponentially. We then show how the decomposed LP formulation can solve a wider class of CTMDP that are quasi decomposable. Indeed it is possible to forbid any combination of sub-actions by adding (possibly many) constraints in the decomposed LP. We prove that, given a set of linear constraints added to the LP, determining whether there exists a deterministic policy solution is NP-complete. We also exhibit simple constraints that allow to forbid some specific combinations of sub-actions. Finally, a numerical study compares computation times of decomposed and non-decomposed formulations for both LP and DP algorithms.

Keywords: Continuous-time Markov decision process; Queueing control; Event-based dynamic programming; Linear programming.

This appendix is based on the article "Linear programming formulations for queueing control problems with action decomposability" (Waserhole *et al.*, 2013a) submitted to Operations Research journal.

A.1 Introduction

Different approaches exist to solve numerically a Continuous-Time Markov Decision Problem (CTMDP) that are based on optimality equations (or Bellman equations). The most popular method is the value iteration algorithm which is essentially a backward Dynamic Programming (DP). Another well known approach is to model a CTMDP as a Linear Programming (LP). LP based algorithms are slower than DP based algorithms. However, LP formulations offer the possibility to add very easily linear constraints on steady state probabilities, which is not the case of DP formulations. Good introductions to CTMDPs can be found in the books of Puterman (1994), Bertsekas (2005b) and Guo and Hernández-Lerma (2009).

In this paper, we consider a special class of CTMDPs that we call action Decomposed CTMDPs (D-CTMPDs). D-CTMDP typically model queueing control problems with several types of events (demand arrival, service end, failure, etc) and where a sub-action (admission, routing, repairing, etc) and also a cost is associated to each type of event. The class of D-CTMPD is related to the concept of event-based DP, first introduced by Koole (1998). Event-based DP is a systematic approach for deriving monotonicity results of optimal policies for various queueing and resource sharing models. Citing Koole: "Event-based DP deals with event operators, which can be seen as building blocks of the value function. Typically it associates an operator with each basic event in the system, such as an arrival at a queue, a service completion, etc. Event-based DP focuses on the underlying properties of the value and cost functions, and allows us to study many models at the same time." The event-based DP framework is strongly related to older works (see e.g. Lippman (1975); Weber and Stidham (1987)).

Apart from the ability to prove structural properties of the optimal policy, the event-based DP framework is also a very natural way to model many queueing control problems. In addition, it allows to reduce drastically the number of actions to be evaluated in the value iteration algorithm. The following example will be used throughout the paper to illustrate our approach and results and will be referred to as the dynamic pricing problem.

Example – Dynamic pricing in a multi-class M/M/1 queue. *Consider a single server with n different classes of clients that are price sensitive (see Figure A.1). There is a finite buffer of size C for each client class. Clients of class $i \in I = \{1, \dots, n\}$ arrive according to an independent Poisson process with rate $\lambda^i(r_i)$ where r_i is a price dynamically chosen in a finite set P of k prices. For clients of class i, the waiting cost per unit of time is b^i and the processing time is exponentially distributed with rate μ^i. At any time the decision maker has to set the entrance price for each class of clients and to decide which class of clients to serve with the objective to maximize the average reward, in the class of preemptive dynamic policies. This problem has been studied, among other works, by Maglaras (2006), Çil et al. (2011), and Li and Neely (2012).*

For this example, the state and action spaces have respectively a cardinality

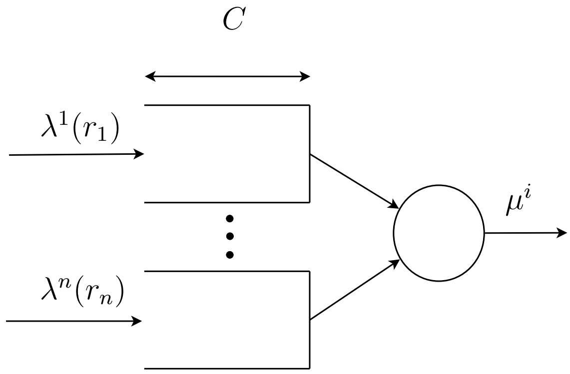


Figure A.1: The multi-class M/M/1 queue with dynamic pricing.

of $C^n + 1$ and nk^n. However, the action selection process in the value iteration algorithm does not require to evaluate the nk^n actions. It is sufficient to evaluate at each iteration only $n(k+1)$ actions (k possibilities for each class of customer and n possibilities for the class to be served). This property has been used intensively in the literature since the seminal paper of Lippman (1975). In the classic LP formulation of the dynamic pricing problem, that one can find in (Puterman, 1994) for instance, the number of variables grows exponentially with the number of possible prices. In this paper, we will show that the LP can be reformulated in a way such that the number of variables grows linearly with the number of possible prices.

Our contributions can be summarized as follows. We first propose a new and natural Quadratic Programming (QP) formulation for CTMDP and relate it to more classic Dynamic Programming (DP) and Linear Programming (LP) formulations. Then, we introduce the class of D-CTMPDs which is probably the largest class of CTMDPs for which the event-based DP approach can be used. We also introduce the class of decomposed randomized policies that will be shown to be dominant among randomized policies. With these new policies, we are able to reformulate the QP and the LP with a number of variables growing linearly with the number of event types. With respect to the decomposed DP, this LP formulation is really simple to write and does not need the uniformization process necessary for the DP formulation which is sometimes source of errors and waste of time. Moreover, it allows to use generic LP related techniques such that sensitivity analysis (Filippi, 2011) or approximate linear programming (Dos Santos Eleutério, 2009).

Another contribution of the paper is to show how to forbid some actions while preserving the structure of the decomposed LP. If some actions (combinations of sub-actions) are forbidden, the DP cannot be decomposed anymore. In the dynamic pricing example, imagine that we want a low price to be selected for at least one class of customer. In the (non-decomposed) DP formulation, it is easy to add this constraint by removing all actions that does not contain a low price. However, it is not possible to decompose anymore the DP, in our opinion. In the decomposed LP formulation, we show how it is possible to remove this action and other combinations of actions by adding simple linear constraints. We also discuss the generic problem of reducing arbitrarily the action space by adding a set of linear constraints in the decomposed LP. Not surprisingly, this problem is difficult and is not appropriate if many actions are removed arbitrarily. When new constraints are added in the decomposed LP, it is also not clear whether deterministic policies remain dominant or not. We even prove that, given a set of linear constraints added to the LP, determining whether there exists a deterministic policy solution is NP-complete. However, for some simple action reductions, we show that deterministic policies remain dominant. We finally present numerical results comparing LP and DP formulations (decomposed or not).

Before presenting the organization of the paper, we mention briefly some related literature addressing MDP with large state space (Bertsekas and Castañon, 1989; Tsitsiklis and Van Roy, 1996) which tries to fight the curse of dimensionality, *i.e.* the exponential growth of the state space size with some parameter of the problem. Another issue, less tackled, appears when the state space is relatively small but the action space is very large. Hu *et al.* (2007) proposes a randomized search method for solving infinite horizon discounted cost discrete-time MDP for uncountable action spaces.

The rest of the paper is organized as follows. We first address the average cost problem. Section A.2 reminds the definition of a CTMDP and formulates it as a QP. We also link the QP formulation with classic LP and DP formulations. In Section A.3, we define properly D-CTMDP and show how the DP and LP can be decomposed for this class of problems. Section A.4 discusses the problem of reducing the action space by adding valid constraints in the decomposed LP. Section A.5 compares numerically computation times of decomposed and non-decomposed formulations for both LP and DP algorithms, for the dynamic pricing problem. Finally, Section A.6 explains how our results can be adapted to a discounted cost criterion.

A.2 Continuous-Time Markov Decision Processes

In this section, we remind some classic results on CTMDPs that will be useful to present our contributions.

A.2.1 Definition

We slightly adapt the definition of a CTMDP given by Guo and Hernández-Lerma (2009). A CTMDP is a stochastic control problem defined by a 5-tuple

$$\left\{S,\ A,\ \lambda_{s,t}(a),\ h_s(a), r_{s,t}(a)\right\}$$

with the following components:

- S is a finite set of states;
- A is a finite set of actions, $A(s)$ are the actions available from state $s \in S$;
- $\lambda_{s,t}(a)$ is the transition rate to go from state s to state t with action $a \in A(s)$;
- $h_s(a)$ is the reward rate while staying in state s with action $a \in A(s)$;
- $r_{s,t}(a)$ is the instant reward to go from state s to state t with action $a \in A(s)$.

Instant rewards $r_{s,t}(a)$ can be included in the reward rates $h_s(a)$ by an easy transformation and reciprocally. Therefore for ease of presentation, we will use the aggregated reward rate $\tilde{h}_s(a) := h_s(a) + \sum_{t\in S} \lambda_{s,t}(a) r_{s,t}(a)$.

We first consider the objective of maximizing the average reward over an infinite horizon, the discounted case will be discussed in Section A.6. We restrict our attention to stationary policies which are dominant for this problem (Bertsekas, 2005b) and define the following policy classes:

- A (*randomized* stationary) *policy* p sets for each state $s \in S$ the probability $p_s(a)$ to select action $a \in A(s)$ with $\sum_{a\in A(s)} p_s(a) = 1$.
- A *deterministic policy* p sets for each state s one action to select: $\forall s \in S$, $\exists a \in A(s)$ such that $p_s(a) = 1$ and $\forall b \in A(s) \setminus \{a\},\ p_s(b) = 0$.
- A *strictly randomized policy* has at least one state where the action is chosen randomly: $\exists s \in S$, $\exists a \in A(s)$ such that $0 < p_s(a) < 1$.

The best average reward policy p^* with gain g^* is solution of the following Quadratic Program (QP) together with a vector ϖ^* (called variant pi" or "pomega"). ϖ_s is to be interpreted as the stationary distribution of state $s \in S$ under policy p.

QP (A.1)

$$
\begin{aligned}
g^* = \max \quad & \sum_{s \in S} \sum_{a \in A(s)} \tilde{h}_s(a) p_s(a) \varpi_s && \text{(A.1a)} \\
\text{s.t.} \quad & \sum_{a \in A(s)} \sum_{t \in S} \lambda_{s,t}(a) p_s(a) \varpi_s = \sum_{t \in S} \sum_{a \in A(t)} \lambda_{t,s}(a) p_t(a) \varpi_t, \quad \forall s \in S, && \text{(A.1b)} \\
& \sum_{s \in S} \varpi_s = 1, && \text{(A.1c)} \\
& \varpi_s \geq 0, && \text{(A.1d)} \\
& \sum_{a \in A(s)} p_s(a) = 1, \quad \forall s \in S, && \text{(A.1e)} \\
& p_s(a) \geq 0. && \text{(A.1f)}
\end{aligned}
$$

QP (A.1) is a natural way to formulate a stationary MDP. It is easy to see that this formulation solves the best average stationary reward policy. First Equations (A.1e) and (A.1f) define the space of admissible stationnary randomized policies p. Secondly for a given policy p, Equations (A.1b)-(A.1d) compute the stationary distribution ϖ of the induced Continuous-Time Markov Chain with gain expressed by Equation (A.1a).

As we will show in Section A.2.3, the usual LP formulation (A.4) can be derived directly from QP (A.1) by simple substitutions of variables. However, in the literature it is classically derived from the linearization of dynamic programming optimality equations given in the following subsection.

A.2.2 Optimality equations

A CTMDP can be transformed into a discrete-time MDP through a uniformization process (Lippman, 1975). In state s, we uniformize the CTMDP with rate $\Lambda_s := \sum_{t \in S} \Lambda_{s,t}$ where $\Lambda_{s,t} := \max_{a \in A(s)} \lambda_{s,t}(a)$. This uniformization rate simplifies greatly the optimality equations and linear programs.

In the rest of the paper, under relatively general conditions (typically for unichain models, see *e.g.* Bertsekas (2005b)) we assume that the optimal average reward g^* is independent of the initial state and is the unique solution together with an associated differential reward vector v^* that satisfies the Bellman's optimality equations:

$$
\frac{g}{\Lambda_s} = T\Big(v(s)\Big) - v(s), \qquad \forall s \in S, \tag{A.2}
$$

with operator T defined as

$$T\Big(v(s)\Big) := \max_{a\in A(s)} \left\{ \frac{1}{\Lambda_s} \left(\tilde{h}_s(a) + \sum_{t\in S} \Big[\lambda_{s,t}(a)v(t) + \Big(\Lambda_{s,t} - \lambda_{s,t}(a)\Big)v(s) \Big] \right) \right\}. \tag{A.3}$$

These optimality equations can be used to compute the best MDP policy. For instance, the value iteration algorithm roughly consists in defining a sequence of value function $v_{n+1} = T(v_n)$ that provides a stationary ϵ-optimal deterministic policy in a number of iterations depending on the desired ϵ.

A.2.3 Linear programming formulation

Since Manne (1960) we know that it is possible to compute the best average reward policy through a LP formulation. From Equations (A.2) and (A.3), one can show that the optimal average reward g^* is the solution of the following program:

$$\begin{aligned} g^* = \min \quad & g \\ \text{s.t.} \quad & \frac{g}{\Lambda_s} \geq T\Big(v(s)\Big) - v(s), && \forall s \in S, \\ & v(s) \in \mathbb{R}, \quad g \in \mathbb{R}. \end{aligned}$$

Linearizing the max function in operator T leads to the following LP formulation and its dual counterpart.

Primal LP

$$\begin{aligned} g^* = \min \quad & g \\ \text{s.t.} \quad & g \geq \tilde{h}_s(a) + \sum_{t\in S} \lambda_{s,t}(a)\Big(v(t) - v(s)\Big), && \forall s \in S,\ \forall a \in A(s), \\ & v(s) \in \mathbb{R}, \quad g \in \mathbb{R}. \end{aligned}$$

Dual LP (A.4)

$$g^* = \max \quad \sum_{s\in S}\sum_{a\in A} \tilde{h}_s(a)\pi_s(a) \tag{A.4a}$$

$$\text{s.t.} \quad \sum_{a\in A(s)}\sum_{t\in S} \lambda_{s,t}(a)\pi_s(a) = \sum_{t\in S}\sum_{a\in A(t)} \lambda_{t,s}(a)\pi_t(a), \qquad \forall s \in S, \tag{A.4b}$$

$$\sum_{s\in S}\sum_{a\in A(s)} \pi_s(a) = 1, \tag{A.4c}$$

$$\pi_s(a) \geq 0. \tag{A.4d}$$

The advantage of the dual formulation is to allow a simple interpretation: variable $\pi_s(a)$ is the average proportion of time spent in state s choosing action a.

A simple way to show that the dual LP (A.4) solves the best average reward policy is to see that it can be obtained from QP (A.1) by the following substitutions of variables:

$$\pi_s(a) = p_s(a)\varpi_s, \quad \forall s \in S,\ \forall a \in A(s).$$

Indeed, any solution (p, ϖ) of QP (A.1) can be mapped into a solution π of dual LP (A.4) with same expected gain thanks to the following mapping:

$$(p, \varpi) \mapsto \pi = \Big(\pi_s(a) = p_s(a)\varpi_s\Big).$$

For the opposite direction, there exists several "equivalent" mappings preserving the gain. Their differences lie in the decisions taken in unreachable states ($\varpi_s = 0$). We exhibit one:

$$\pi \mapsto (p, \varpi) = \left(p_s(a) = \left\{ \begin{array}{ll} \frac{\pi_s(a)}{\varpi_s}, & \text{if } \varpi_s \neq 0 \\ 1, & \text{if } \varpi_s = 0 \text{ and } a = a^1 \\ 0, & \text{otherwise} \end{array} \right\}, \varpi_s = \sum_{a \in A(s)} \pi_s(a) \right).$$

Since any solution of QP (A.1) can be mapped to a solution of dual LP (A.4) and conversely, in the sequel we overload the word *policy* as follows:

- We (abusively) call *(randomized) policy* a solution π of the dual LP (A.4).
- We say that π is a *deterministic policy* if it satisfies $\pi_s(a) \in \{0, \varpi_s = \sum_{a \in A(s)} \pi_s(a)\}$, $\forall s \in S,\ \forall a \in A(s)$.

A.3 Action Decomposed Continuous-Time Markov Decision Processes

A.3.1 Definition

An action Decomposed CTMDP (D-CTMDP) is a CTMDP such that:

- In each state $s \in S$, the action space can be written as the Cartesian product of $n_s \geq 1$ sub-action sets: $A(s) = A_1(s) \times \ldots \times A_{n_s}(s)$. An action $a \in A(s)$ is then composed by n_s sub-actions $(a_1, \ldots, a_{n_s})$ where $a_i \in A_i(s)$.
- *Sub-action* a_i increases the transition rate from s to t by $\lambda^i_{s,t}(a_i)$, the reward rate by $h^i_s(a_i)$ and the instant reward rate by $r^i_{s,t}(a_i)$.
  - The resulting transition rate from s to t is then $\lambda_{s,t}(a) = \sum_{i=1}^{n_s} \lambda^i_{s,t}(a_i)$.

- The resulting aggregated reward rate in state s when action a is taken is then $\tilde{h}_s(a) = \sum_{i=1}^{n_s} \tilde{h}_s^i(a_i)$ with $\tilde{h}_s^i(a_i) = h_s(a_i) + \sum_{t \in S} \lambda_{s,t}^i(a_i) r_{s,t}^i(a_i)$.

D-CTMDPs typically model queueing control problems with several types of events (demand arrival, service end, failure, etc), an action associated to each type of event (admission control, routing, repairing, etc) and also a cost associated to each type of event. Event-based DP, as defined by Koole (1998), is included in the class of D-CTMDPs.

For ease of notation, we assume without loss of generality that each state $s \in S$ has exactly $n_s = n$ independent sub-action sets, with $I = \{1, \cdots, n\}$, and that each sub-action set $A_i(s)$ contains exactly k sub-actions.

We introduce the concept of decomposed policy.

- A *(randomized) decomposed policy* is a vector $\mathring{p} = ((\mathring{p}_s^1, \ldots, \mathring{p}_s^n),\ s \in S)$ such that for each state s there is a probability $\mathring{p}_s^i(a_i)$ to select sub-action $a_i \in A_i(s)$ with $\sum_{a_i \in A_i(s)} \mathring{p}_s^i(a_i) = 1,\ \forall s \in S,\ \forall i \in I$. The probability to choose action $a = (a_1, \cdots, a_n)$ in state s is then $p_s(a) = \prod_{i \in I} \mathring{p}_s^i(a_i)$.
- A decomposed policy $\mathring{p}$ is said *deterministic* if $\forall s \in S,\ \forall i \in I,\ \exists a_i \in A_i(s)$ such that $\mathring{p}_s^i(a_i) = 1$ and $\forall b_i \in A_i(s) \setminus \{a_i\},\ \mathring{p}_s^i(b_i) = 0$. In other words, $\mathring{p}$ selects one sub-action for each state s and each sub-action set A_i.

In the following we will see that decomposed policies are dominant for D-CTMDPs. It is interesting since a decomposed policy $\mathring{p}$ is described in a much more compact way than a classic policy p.

Simply applying the definition, one can check that the best average reward decomposed policy $\mathring{p}^*$ is solution of the following quadratic program where $\mathring{\varpi}_s$ is to be interpreted as the stationary distribution of state $s \in S$.

Decomposed QP (A.5)

$$g^* = \max \quad \sum_{s\in S}\sum_{i\in I}\sum_{a_i\in A_i(s)} \tilde{h}^i_s(a_i)\mathring{p}^i_s(a_i)\mathring{\varpi}_s \tag{A.5a}$$

$$\text{s.t.} \quad \sum_{i\in I}\sum_{a_i\in A_i(s)}\sum_{t\in S}\lambda^i_{s,t}(a_i)\mathring{p}^i_s(a_i)\mathring{\varpi}_s = \sum_{t\in S}\sum_{i\in I}\sum_{a_i\in A_i(t)} \lambda^i_{t,s}(a_i)\mathring{p}^i_t(a_i)\mathring{\varpi}_t, \qquad \forall s\in S, \tag{A.5b}$$

$$\sum_{s\in S} \mathring{\varpi}_s = 1, \tag{A.5c}$$

$$\mathring{\varpi}_s \geq 0, \tag{A.5d}$$

$$\sum_{a_i\in A_i(s)} \mathring{p}^i_s(a_i) = 1, \qquad \forall s\in S,\ \forall i\in I, \tag{A.5e}$$

$$\mathring{p}^i_s(a_i) \geq 0. \tag{A.5f}$$

Example – Dynamic pricing in a multi-class M/M/1 queue (D-CTMDP formulation)**.** *We continue the example started in Section A.1. This problem can be modeled as a D-CTMDP with state space $S = \{(s_1,\dots,s_n) \mid s_i \leq C,\ \forall i\in I\}$. In each state $s\in S$, there is $n_s = (n+1)$ sub-actions and an action can be written as $a = (r_1,\dots,r_n,d)$ with r_i the price decided to be offered to client class i and d the client class to process. The action space is then $A = P^n \times D$ with $D = \{1,\dots,n\}$. The waiting cost in state $(s_1,\dots,s_n)$ is independent of the action selected and is worth $\sum_i h^i s_i$. The reward rate incurred by sub-action r_i is $\lambda_i(r_i)r^i$. Let the function $\mathbb{1}_b$ equals 1 if boolean expression b is worth true and 0 otherwise. The resulting aggregated reward rate in state $s = (s_1,\dots,s_n)$ when action $a = (r_1,\dots,r_n,d)$ is selected is then $\tilde{h}_s(a) = \sum_{i=1}^n \tilde{h}^i_s(r_i)$ with $\tilde{h}^i_s(r_i) = h^i s_i + \mathbb{1}_{s_i<C}\lambda_i(r_i)r^i$.*

For this example, the cardinality of the state space and the action space are respectively $|S| = (C+1)^n$ and $|A| = k^n n$.

A.3.2 Optimality equations

Optimality equations for CTMDPs can be rewritten in the context of a D-CTMDPs to take advantage of decomposition properties. Let $\Lambda^i_{s,t} = \max_{a_i\in A_i(s)} \lambda^i_{s,t}(a_i)$. The uniformization rate is again $\Lambda_s = \sum_{t\in S}\Lambda_{s,t}$ where $\Lambda_{s,t}$, as defined previous section, can be rewritten as follows:

$$\Lambda_{s,t} = \max_{a\in A(s)} \lambda_{s,t}(a) = \max_{\substack{(a_1,\ \dots,\ a_n)\\ \in\ A_1(s)\times\dots\times A_n(s)}} \sum_{i\in I}\lambda^i_{s,t}(a_i) = \sum_{i\in I}\max_{a_i\in A_i(s)}\lambda^i_{s,t}(a_i) = \sum_{i\in I}\Lambda^i_{s,t}.$$

Operator T as defined in Equation (A.3) can be rewritten as:

$$T\Big(v(s)\Big) = \max_{\substack{(a_1,\ \dots,\ a_n)\\ \in\ A_1(s)\times\dots\times A_n(s)}} \left\{ \frac{1}{\Lambda_s} \sum_{i\in I} \left(\tilde{h}^i_s(a_i) + \sum_{t\in S} \left[\lambda^i_{s,t}(a_i)v(t) + \Big(\Lambda^i_{s,t} - \lambda^i_{s,t}(a_i)\Big)v(s) \right] \right) \right\}. \tag{A.6}$$

That we can decompose as:

$$T\Big(v(s)\Big) = \frac{1}{\Lambda_s} \sum_{i\in I} \left(\max_{a_i\in A_i(s)} \left\{ \tilde{h}^i_s(a_i) + \sum_{t\in S} \left[\lambda^i_{s,t}(a_i)v(t) + \Big(\Lambda^i_{s,t} - \lambda^i_{s,t}(a_i)\Big)v(s) \right] \right\} \right). \tag{A.7}$$

The value iteration algorithm is much more efficient if T is expressed as in the latter equation. Experimental results presented in Section A.5 show it clearly. Indeed computing the maximum requires n^k evaluations in Equation (A.6) and nk in Equation (A.7). To the best of our knowledge, this decomposition property of operator T is used in many queueing control problems (see Koole (1998) and related papers) but has not been formalized as generally as in this paper.

Example – Dynamic pricing in a multi-class M/M/1 queue (DP approach)**.** *We can now write down the optimality equations. We use the following uniformization: let $\Lambda = \sum_{i\in I} \Lambda^i + \Delta$ with $\Lambda^i = \max_{r_i\in P}\{\lambda^i(r_i)\}$ and $\Delta = \max_{i\in I}\{\mu^i\}$.*

For a state $s = (s_1, \dots, s_n)$ and with e_i the unitvector of the i^{th} coordinate, the operator T for classic optimality equations can be defined as follows:

$$\begin{aligned} T\Big(v(s)\Big) = \max_{(r_1,\dots,r_n,d)\in A} \Bigg\{ & \sum_{i\in I} \left[\tilde{h}^i(r_i) + \mathbb{1}_{s_i<C}\lambda^i(r_i)v(s+e_i) \right] \\ & + \mathbb{1}_{s_d>0}\mu^d v(s-e_d) + \left(\Lambda - \sum_{i\in I} \mathbb{1}_{s_i<C}\lambda^i(r_i) + \Delta - \mathbb{1}_{s_d>0}\mu^d \right) v(s) \Bigg\}. \end{aligned}$$

Since we are dealing with a D-CTMDP, operator T can also be decomposed as:

$$\begin{aligned} T\Big(v(s)\Big) = \frac{1}{\Lambda} \Bigg(& \sum_{i\in I} \left[\tilde{h}^i(r_i) + \max_{\substack{r_i\in P\\ s_i<C}} \left\{ \lambda^i(r_i)v(s+e_i) + \Big(\Lambda^i - \lambda^i(r_i)\Big)v(s) \right\} \right] \\ & + \max_{\substack{d\in D\\ s_d>0}} \left\{ \mu^d v(s-e_d) + (\Delta - \mu^d)v(s) \right\} \Bigg). \end{aligned}$$

A.3.3 LP formulation

Let $\mathring{\pi}^i_s(a_i)$ be interpreted as the average proportion of time spent in state s choosing action $a_i \in A_i(s)$ among all sub-actions $A_i(s)$. From decomposed QP (A.5) we can build the LP (A.8) formulation with simple substitutions of variable:

$$\mathring{\pi}^i_s(a_i) = \mathring{\pi}^i_s(a_i)\mathring{\varpi}_s, \quad \forall s \in S,\ \forall i \in I,\ \forall a_i \in A_i(s).$$

We obtain that g^* is the solution of the following LP formulation:

Decomposed Dual LP (A.8)

$$g^* = \max \quad \sum_{s\in S}\sum_{i\in I}\sum_{a_i\in A_i(s)} \tilde{h}^i_s(a_i)\mathring{\pi}^i_s(a_i) \tag{A.8a}$$

$$\text{s.t.} \quad \sum_{i\in I}\sum_{a_i\in A_i(s)}\sum_{t\in S}\lambda^i_{s,t}(a_i)\mathring{\pi}^i_s(a_i) = \sum_{t\in S}\sum_{i\in I}\sum_{a_i\in A_i(t)} \lambda^i_{t,s}(a_i)\mathring{\pi}^i_t(a_i), \qquad \forall s \in S, \tag{A.8b}$$

$$\sum_{a_i\in A_i(s)} \mathring{\pi}^i_s(a_i) = \mathring{\varpi}_s, \qquad \forall s \in S,\ \forall i \in I, \tag{A.8c}$$

$$\sum_{s\in S} \mathring{\varpi}_s = 1, \tag{A.8d}$$

$$\mathring{\pi}^i_s(a_i) \geq 0, \quad \mathring{\varpi}_s \geq 0. \tag{A.8e}$$

The decomposed dual LP formulation (A.8) has $|S|(kn+1)$ variables and $|S|((k+1)n+2)+1$ constraints. It is much less than the classic dual LP formulation (A.4) that has $|S|k^n$ variables and $|S|(k^n+1)$ constraints.

Lemma 10. *Any solution $(\mathring{p}, \mathring{\varpi})$ of the decomposed QP (A.5) can be mapped into a solution $(\mathring{\pi}, \mathring{\varpi})$ of the decomposed dual LP (A.8) with same expected gain thanks to the following mapping:*

$$(\mathring{p}, \mathring{\varpi}) \to (\mathring{\pi}, \mathring{\varpi}) = \left(\mathring{\pi}^i_s(a_i) = \mathring{\pi}^i_s(a_i)\mathring{\varpi}_s,\ \mathring{\varpi}\right).$$

For the opposite direction, there exists several "equivalent" mappings preserving the gain. Their differences lie in the decisions taken in unreachable states ($\mathring{\varpi}_s = 0$). We exhibit one:

$$(\mathring{\pi}, \mathring{\varpi}) \mapsto (\mathring{p}, \mathring{\varpi}) = \left(\mathring{p}^i_s(a_i) = \left\{\begin{array}{ll} \frac{\mathring{\pi}^i_s(a_i)}{\mathring{\varpi}_s}, & \textit{if } \mathring{\varpi}_s \neq 0 \\ 1, & \textit{if } \mathring{\varpi}_s = 0 \textit{ and } a_i = a_1 \\ 0, & \textit{otherwise} \end{array}\right\},\ \mathring{\varpi}\right).$$

$\square$

Since any solution of the decomposed QP (A.5) can be matched to a solution of decomposed dual LP (A.8) and conversely (Lemma 10), in the sequel we again overload the word *policy* as follows:

- We (abusively) call *(randomized) decomposed policy* a solution $(\mathring{\varpi}, \mathring{\pi})$ of the decomposed dual LP (A.8).
- We say that $(\mathring{\varpi}, \mathring{\pi})$ is a *deterministic policy* if it satisfies $\mathring{\pi}_s^i(a_i) \in \{0, \mathring{\varpi}_s\}$, $\forall s \in S,\ \forall i \in I,\ \forall a_i \in A_i(s)$.

Dualizing the decomposed dual LP (A.8), we obtain the following primal version:

Decomposed Primal LP (A.9)

$$
\begin{aligned}
g^* = \min \quad & g \\
\text{s.t.} \quad & m(s,i) \geq \tilde{h}_s^i(a_i) + \sum_{t \in S} \lambda_{s,t}^i(a_i)\Big(v(t) - v(s)\Big), \quad \forall s \in S,\ \forall i \in I,\ \forall a_i \in A_i(s), \\
& g \geq \sum_{i \in I} m(s,i), \qquad \forall s \in S, \\
& m(s,i) \in \mathbb{R}, \quad v(s) \in \mathbb{R}, \quad g \in \mathbb{R}.
\end{aligned}
$$

Note that the decomposed primal LP (A.9) could have also been obtained using the optimality equations (A.7). Indeed, under some general conditions (Bertsekas, 2005b), the optimal average reward g^* is independent from the initial state and together with an associated differential cost vector v^* it satisfies the optimality equations (A.7). The optimal average reward g^* is hence the solution of the following equations:

$$
\begin{aligned}
g^* = \min \quad & g \\
\text{s.t.} \quad & \frac{g}{\Lambda_s} \geq T\Big(v(s)\Big) - v(s), \qquad \forall s \in S.
\end{aligned}
$$

That can be reformulated using decomposability to have:

$$
\begin{aligned}
g^* &= \max_{s \in S} \left\{ \Lambda_s \Big(T(v(s)) - v(s) \Big) \right\} \\
&= \max_{s \in S} \left\{ \sum_{i \in I} \left(\max_{a_i \in A_i(s)} \left\{ \tilde{h}_s^i(a_i) + \sum_{t \in S} \left[\lambda_{s,t}^i(a_i) v(t) + \Big(\Lambda_{s,t}^i - \lambda_{s,t}^i(a_i) \Big) v(s) \right] - \Lambda_s^i v(s) \right\} \right) \right\} \\
&= \max_{s \in S} \left\{ \sum_{i \in I} \left(\max_{a_i \in A_i(s)} \left\{ \tilde{h}_s^i(a_i) + \sum_{t \in S} \lambda_{s,t}^i(a_i) \Big(v(t) - v(s) \Big) \right\} \right) \right\}. \qquad \text{(A.10)}
\end{aligned}
$$

The LP (A.9) can also be obtained from Equation (A.10) using the following lemma.

Lemma 11. *For any finite sets $S,\ I,\ A$ and any data coefficients $\gamma_{s,i,a} \in \mathbb{R}$ with $s \in S,\ i \in I$ and $a \in A$, the value*

$$
g^* = \max_{s \in S} \left\{ \sum_{i \in I} \max_{a \in A} \left\{ \gamma_{s,i,a} \right\} \right\}
$$

is the solution of the following LP:

$$
\begin{aligned}
g^* = \min \quad & g \\
s.t. \quad & m(s,i) \geq \gamma_{s,i,a}, && \forall s \in S,\ \forall i \in I,\ \forall a \in A, \\
& g \geq \sum_{i \in I} m(s,i), && \forall s \in S, \\
& m(s,i) \in \mathbb{R}, && \forall s \in S,\ \forall i \in I, \\
& g \in \mathbb{R}.
\end{aligned}
$$

Proof. Let g^* be an optimal solution of this LP. First it is trivial that $g^* \geq \max_{s \in S} \left\{ \sum_{i \in I} m(s,i) \right\}$ and that Moreover we are minimizing g without any other constraints, hence $g^* = \max_{s \in S} \left\{ \sum_{i \in I} m(s,i) \right\}$. Secondly for any optimal solution g^*, there exists $s' \in S$ such that $\sum_{i \in I} m(s',i) = g^*$ and $\forall i \in I,\ m(s',i) = \max_{a \in A} \{\gamma_{s',i,a}\}$, otherwise there would exist a strictly better solution. Therefore finally $g^* = \max_{s \in S} \left\{ \sum_{i \in I} \max_{a \in A} \{\gamma_{s,i,a}\} \right\}$. □

Example – Dynamic pricing in a multi-class M/M/1 queue (LP approach)**.** *With $a = (r_1, \ldots, r_n, d) \in A$, we can now formulate its classic dual LP formulation:*

$$
\begin{aligned}
\max \ & \sum_{s \in S} \sum_{a \in A} \left(\sum_{i=1}^{n} \tilde{h}^i(r_i) \right) \pi_s(a) \\
s.t. \ & \sum_{a \in A} \left(\mathbb{1}_{s_d > 0} \mu^d + \sum_{i=1}^{n} \mathbb{1}_{s_i < C} \lambda^i(r_i) \right) \pi_s(a) \\
& \quad = \sum_{a \in A} \left(\sum_{i=1}^{n} \mathbb{1}_{s_i > 0} \lambda^i(r_i) \pi_{s-e_i}(a) + \mathbb{1}_{s_d < C} \mu^d \pi_{s+e_d}(a) \right), \qquad \forall s \in S, \\
& \sum_{s \in S} \sum_{a \in A} \pi_s(a) = 1, \\
& \mathring{\pi}_s(a) \geq 0.
\end{aligned}
$$

And its decomposed Dual LP formulation:

$$
\begin{aligned}
\max \quad & \sum_{s\in S}\sum_{i=1}^{n} \tilde{h}^i(r_i)\mathring{\pi}_s^i(r_i) \\
s.t. \quad & = \sum_{i\in I}\sum_{r_i\in P} \mathbb{1}_{s_i>0}\lambda^i(r_i)\mathring{\pi}^i_{s-e_i}(r_i) + \sum_{d\in D}\mathbb{1}_{s_d<C}\mu^d\mathring{\pi}_{s+e_d}(d), && \forall s\in S,\\
& \sum_{r_i\in P} \mathring{\pi}^i_s(r_i) = \mathring{\varpi}_s, && \forall s\in S,\ \forall i\in I,\\
& \sum_{d\in D} \mathring{\pi}_s(d) = \mathring{\varpi}_s, && \forall s\in S,\\
& \sum_{s\in S} \mathring{\varpi}_s = 1, \\
& \mathring{\pi}^i_s(r_i)\geq 0, \quad \mathring{\pi}_s(d)\geq 0, \quad \mathring{\varpi}_s \geq 0.
\end{aligned}
$$

A.3.4 Polyhedral results

Seeing the decomposed dual LP (A.8) as a reformulation of the decomposed QP (A.5), see Lemma 10, it is clear that it gives a policy maximizing the average reward. However, it doesn't provide any structure of optimal solutions. For this purpose, Lemma 12 gives two mappings linking classic and decomposed policies that are used in Theorem 11 to prove polyhedral results showing the dominance of deterministic policies. This means that the simplex algorithm on the decomposed dual LP (A.8) will return the best average reward deterministic policy.

Lemma 12. *Let $a(i)$ be the i^{th} coordinate of vector a. The following policy mappings preserve the strictly randomized and deterministic properties:*

$$
D:\ p\mapsto \mathring{p} = \left(\mathring{p}^i_s(a_i) = \sum_{a\in A(s)/a(i)=a_i} p_s(a)\right); \qquad D^{-1}:\ \mathring{p}\mapsto p = \left(p_s(a_1,\dots,a_n) = \prod_{i\in I}\mathring{p}^i_s(a_i)\right).
$$

Moreover:
(a) D is linear.
(b) The following policy transformations preserve moreover the excepted gain:

1. $(p,\varpi)\mapsto(\mathring{p},\mathring{\varpi}) = (D(p),\varpi)$;
2. $\pi\mapsto(\mathring{\pi},\mathring{\varpi}) = (D(\pi), \mathring{\varpi}_s = \sum_{a\in A(s)}\pi_s(a))$;
3. $(\mathring{p},\mathring{\varpi})\mapsto(p,\varpi) = (D^{-1}(\mathring{p}),\mathring{\varpi})$;
4. $(\mathring{\pi},\mathring{\varpi})\mapsto(\pi,\varpi) = (D^{-1}(\mathring{\pi}),\mathring{\varpi})$.

□

Theorem 11. *The best decomposed CTMDP average reward policy is solution of the decomposed dual LP (A.8). Equations (A.8b)-(A.8e) describe the convex hull of deterministic policies.*

Proof. We call P the polytope defined by constraints (A.8b)-(A.8e). From Lemma 10 we know that all policies are in P. To prove that vertices of P are deterministic policies we use the characterization that a vertex of a polytope is the unique optimal solution for some objective.

Assume that $(\mathring{\pi}, \mathring{\varpi})$ is a strictly randomized decomposed policy, optimal solution with gain $\mathring{g}$ of the decomposed dual LP (A.8) for some objective $\tilde{h}_o$. From Lemma 12 we know that there exists a strictly randomized non decomposed policy (π, ϖ) with same expected gain. Deterministic policies are dominant in non decomposed models, therefore there exists a deterministic policy (π^*, ϖ^*) with gain $g^* \geq \mathring{g}$. From Lemma 12 we can convert (π^*, ϖ^*) into a deterministic decomposed policy $(\mathring{\pi}^*, \mathring{\varpi}^*)$ with same expected gain $\mathring{g}^* = g^* \geq \mathring{g}$. Since $(\mathring{\pi}, \mathring{\varpi})$ is optimal we have then $\mathring{g}^* = g^*$ which means that $(\mathring{\pi}, \mathring{\varpi})$ is not the unique optimal solution for objective $\tilde{h}_o$. Therefore a strictly decomposed randomized policy can't be a vertex of the decomposed LP (A.8) and P is the convex hull of deterministic policies. □

A.3.5 Benefits of decomposed LP

First, recall that with the use of action decomposability, the decomposed LP (A.8) allows to have a complexity polynomial in the number of independent sub-action sets: $|S|(kn + 1)$ variables and $|S|((k + 1)n + 2)$ constraints for the dual whereas in the classic it grows exponentially: $|S|k^n$ variables and $|S|(k^n + 1)$ constraints. In Section A.5 we will see that it has a substantial impact on the computation time.

Secondly, even if the LP (A.8) is slower to solve than DP (A.7), as shown experimentally in Section A.5, this mathematical programming approach offers some advantages. First, LP formulations can help to characterize the polyhedral structure of discrete optimization problems, see Büyüktahtakin (2011). Secondly, there is in the LP literature generic methods directly applicable such as sensitive analysis, see Filippi (2011), or approximate linear programming techniques, see Dos Santos Eleutério (2009). Another interesting advantage is that the dual LP (A.8) is really simple to write and does not need the uniformization necessary to the DP (A.7) which is sometimes source of waste of time and errors.

Finally, a big benefit of the LP formulation is the ability to add extra constraints that are not known possible to consider in the DP. A classic constraint that is known possible to add only in the LP formulation is to restrict the stationary distribution on a subset $T \subset S$ of states to be greater than a parameter q, for instance to force a quality of service:

$$\sum_{s \in T} \varpi_s \geq q.$$

Nevertheless, we have to be aware that such constraint can enforce strictly randomized policies as optimal solutions. The constraints discussed in the next section preserve the dominance of deterministic policies.

A.4 Decomposed LP for a broader class of large action space MDP

A.4.1 On reducing action space and preserving decomposability

In this section, we use the decomposed LP formulation to solve polynomially in the number of sub-action sets a broader class of MDP with large action space. We tackle CTMDPs that have a decomposable action space except in some state $s \in S$ where some actions $a^f = \{a_1, \dots, a_n\} \in A(s) = \prod_{i=1}^n A_i(s),\ f \in F$ are forbidden. Their action space $A'(s) = A(s) \setminus \{a^f,\ f \in F\} \subset A(s)$ is not decomposable anymore, although it has a special structure. Hence, event-based DP techniques are not applicable to solve the best policy. The decomposed LP (A.8) is also useless as it is. However, we can use polyhedral properties to model an action space reduction in the LP. In Theorem 13, we show that it is possible to reduce the action space of any state $s \in S$ to $A'(s) \subseteq A(s)$, while preserving the action decomposability benefits. It can be done by adding a set of constraints to the decomposed dual LP (A.8). In Corollary 5, we provide a *state-policy decomposition criteria* to verify if a set of constraints correctly models an action space reduction. It is a sufficient condition, remain to find such set of constraints.

The QP (A.5) has the advantage of considering explicitly the decision variables: in a state s, $\mathring{p}^i_s(a_i)$ is the discrete probability to choose sub-action $a_i \in A_i(s)$. Hence, adding constraints on variables $\mathring{p}$ drives the *average behavior* of the system. Yet, QP (A.5) is hard to solve as it is, we prefer to solve the decomposed dual LP (A.8). To include QP (A.5) constraints in the decomposed dual LP (A.8), recall the substitution of variables: $\mathring{p}^i_s(a_i) = \frac{\mathring{\pi}^i_s(a_i)}{\mathring{\varpi}_s}$. We define now a general constraint on variables $\mathring{p}$, that remains linear in the decomposed dual LP (A.8) after the substitution of variables.

Definition 9 (Action reduction constraint)**.** *Let $s \in S$ be a state, R be a set of sub-actions available in s, m and M be two integers. An action reduction constraint (s, R, m, M) forces to select in average in state s at least m and at most M sub-actions a_i out of the set R. The following equation defines the space of feasible policies:*

$$m \leq \sum_{a_i \in R} \mathring{p}^i_s(a_i) = \sum_{a_i \in R} \frac{\mathring{\pi}^i_s(a_i)}{\mathring{\varpi}_s} \leq M. \tag{A.11}$$

Example – Dynamic pricing in a multi-class M/M/1 queue (Adding extra constraints in average)**.** *Say we have two prices* high h *and* low l *for the n classes of clients. In a state s we have then the following set of actions: $A(s) = P(s) \times D(s)$ where $P(s) = \prod_{i=1}^n P_i(s)$ and $P_i(s) = \{h_i, l_i\}$. Assume that, for some marketing reasons, at least one low price needs to be offered (selected). In the non decomposed model, this constraint is easily expressible by a new space of action $P'(s) = P(s) \setminus \{(h_1, \dots, h_n)\}$ removing the action where all high prices are selected. However, with this new action space it is not possible to decompose this MDP anymore, even though there is still some structure in the problem.*

We can use action reduction constraints to forbid solutions with only high prices by selecting in average: at most $n-1$ high prices (sub-action h_i) as in Equation (A.12a), or at least one low price (sub-action l_i) as in Equation (A.12b):

$$\sum_{i=1}^{n} \mathring{p}^i_s(h_i) = \sum_{i=1}^{n} \frac{\mathring{\pi}^i_s(h_i)}{\mathring{\varpi}_s} \leq n-1, \tag{A.12a}$$

$$\sum_{i=1}^{n} \mathring{p}^i_s(l_i) = \sum_{i=1}^{n} \frac{\mathring{\pi}^i_s(l_i)}{\mathring{\varpi}_s} \geq 1. \tag{A.12b}$$

Now, if we want now to select exactly $n/2$ high prices, $A'(s) = \{(a_1, \dots, a_n) \mid \sum_{i=1}^n \mathbb{1}_{a_i = l} = n/2\}$, the number of actions to remove from the original action space is exponential in n. However, there is a simple way to model this constraint in average with an action reduction constraint:

$$\sum_{i=1}^{n} \mathring{p}^i_s(h_i) = \sum_{i=1}^{n} \frac{\mathring{\pi}^i_s(h_i)}{\mathring{\varpi}_s} = \frac{n}{2}. \tag{A.13}$$

An action combination constraint drives the average behavior of the system. Yet, together with decomposed dual LP (A.8) we do not know whether it provides optimal deterministic policies. Theorem 13 proves the existence of a set of constraints correctly modeling any action space reduction. However, the number of constraints necessary to model it might be an issue, the decomposed formulation might become less efficient than the non-decomposed one (Proposition 11). One might conjecture a "valid" set of constraints and Corollary 5 gives a sufficient condition to check them. However, applying Corollary 5 involves solving a co-NP complete problem (Proposition 12). And given a set of constraints, it is even NP-complete to check whether there exists one feasible deterministic policy (Proposition 13).

Although it is hard in general to prove that a given set of linear equations models correctly a reduced action space, it is nevertheless possible to exhibit some valid constraints. The following theorem (consequence of Corollary 5) states that we can use several action reduction constraints at the same time (under some assumptions) and preserving dominance of deterministic policies.

Theorem 12 (Combination of action reduction constraints)**.** *For a set of action reduction constraints $\{(s_j, R_j, m_j, M_j) \mid j \in J\}$, where no sub-action a_i is present in more than one action reduction constraint R_j, i.e. $\bigcap_{j \in J} R_j = \emptyset$, the decomposed dual LP (A.8) together with Equations $\{(\text{A.11}) \mid j \in J\}$ preserves the dominance of deterministic policies. Moreover, the solution space of this LP is the convex hull of deterministic policies respecting the action reduction constraints.*

The proof of this theorem is given in Section A.4.2. Applying this theorem to our example, we can verify that Equations (A.12a), (A.12b) or (A.13) correctly model an action space reduction.

A.4.2 State policy decomposition criteria

In the following, for each $s \in S$, $\mathring{\pi}_s$ (resp. $\mathring{p}_s$) represents the matrix of variables $\mathring{\pi}_s^i(a_i)$ (resp. $\mathring{p}_s^i(a_i)$) with $i \in I$ and $a_i \in A_i(s)$. The next theorem states that there exists a set of constraints to add to the decomposed dual LP (A.8) so that it correctly solves the policy in A' maximizing the average reward criterion and that the maximum is attained by a deterministic policy.

Theorem 13. *For a decomposed CTMDP with a reduced action space $A'(s) \subseteq A(s)$, $\forall s \in S$, there exists a set of linear constraints $\{\mathring{B}_s \mathring{p}_s \leq \mathring{b}_s,\ \forall s \in S\}$ that describes the convex hull of deterministic decomposed policies $\mathring{p}$ in A'. Moreover $\{\mathring{B}_s \mathring{\pi}_s \leq \mathring{b}_s \mathring{\varpi}_s,\ \forall s \in S\}$ together with equations (A.8b)-(A.8e) defines the convex hull of decomposed deterministic policies $(\mathring{\pi}, \mathring{\varpi})$ in A'.*

Proof. Equations (A.1e) and (A.1f) of the (non decomposed) QP (A.1) specify the space of feasible policies p for a classic CTMDP. For each state $s \in S$ we can redefined this space as the convex hull of all feasible deterministic policies: $p_s \in \text{conv}\{p_s \mid \exists a \in A'(s) \text{ s.t. } p_s(a) = 1\}$. The mapping D defined in Lemma 12 is linear. Note that for any linear mapping M and any finite set X, $\text{conv}(M(X)) = M(\text{conv}(X))$. Hence for each state $s \in S$ the convex hull H_s of CTMDPs policies with support in $A'(s)$ is mapped to the convex hull $\mathring{H}_s$ of decomposed CTMDPs state policy in $A'(s)$:

$$\begin{aligned} D(H_s) &= \mathring{H}_s \\ \Leftrightarrow\ D\Big(\operatorname{conv}\Big\{p_s \mid \exists a \in A'(s) \text{ s.t. } p_s(a) = 1\Big\}\Big) &= \operatorname{conv}\Big\{D(p_s) \mid \exists a \in A'(s) \text{ s.t. } p_s(a) = 1\Big\} \\ &= \operatorname{conv}\Big\{\mathring{p}_s \mid \exists (a_1, \ldots, a_n) \in A'(s) \text{ s.t. } \mathring{p}^i_s(a_i) = 1\Big\}. \end{aligned}$$

Recall (a particular case of) Minkowski-Weyl's theorem: for any finite set of vectors $A' \subseteq \mathbb{R}^n$ there exists a finite set of linear constraints $\{Bv \leq b\}$ that describes the convex hull of vectors v in A'. The set $\mathring{H}_s$ is the convex hull of a finite set, hence from Minkowski-Weyl's theorem there exists a matrix $\mathring{B}_s$ and a vector $\mathring{b}_s$ such that $\mathring{H}_s$ is the set of vectors $\mathring{p}_s$ satisfying the constraints $\mathring{B}_s\mathring{p}_s \leq \mathring{b}_s$. We deduce that replacing Equations (A.5e) and (A.5f) (convex hull of policies in A) by constraints $\{\mathring{B}_s\mathring{p}_s \leq \mathring{b}_s,\ \forall s \in S\}$ (convex hull of policies in A') in the decomposed dual QP (A.5) solves the optimal average reward policy in A'.

With substitutions of variables, one derives the constraints $C := \{\mathring{B}_s\mathring{\pi}_s \leq \mathring{b}_s\mathring{\varpi}_s,\ \forall s \in S\}$ which are linear in $(\mathring{\pi}_s, \mathring{\varpi}_s)$. The decomposed dual LP (A.8) together with constraints C hence solves the optimal average reward policies in A'. However, at this stage we do not know yet if the vertices of the polytope defined by Equations (A.8b)-(A.8e) together with constraints C are deterministic policies. To prove it, as in Theorem 11, we use the characterization that a vertex of a polytope is the unique optimum solution for some objective. Assume that $(\mathring{\pi}, \mathring{\varpi})$ is a strictly randomized decomposed policy of gain $\mathring{g}$, optimal solution of the decomposed dual LP (A.8) together with constraints C for some objective $\tilde{h}_0$. From Lemma 12, policy $(\mathring{\pi}, \mathring{\varpi})$ can be mapped to a strictly randomized non decomposed policy (π, ϖ) in the convex hull of A', with expected gain $\mathring{g}$, that is dominated by a deterministic policy $(\pi^*, \varpi^*) \in A'$ with gain $g^* \geq \mathring{g}$. Policy (π^*, ϖ^*) can be again mapped to a deterministic decomposed policy $(\mathring{\pi}^*, \mathring{\varpi}^*) \in A'$ with same expected gain $\mathring{g}^* = g^* \geq \mathring{g}$. But since policy $(\mathring{\pi}, \mathring{\varpi})$ is optimal we have $\mathring{g}^* = g^*$, which means that $(\mathring{\pi}, \mathring{\varpi})$ is not the unique optimal solution for objective $\tilde{h}_0$. Therefore, a strictly decomposed randomized policy can't be a vertex of the decomposed dual LP (A.8) together with constraints C. □

Corollary 5 (State policy decomposition criteria)**.** *If the vertices of the polytope $\{\mathring{B}_s\mathring{p}_s \leq \mathring{b}_s\}$ are $\{0,1\}$-vectors for each state $s \in S$, the decomposed dual LP (A.8) together with constraints $\{\mathring{B}_s\mathring{\pi}_s \leq \mathring{b}_s\mathring{\varpi}_s,\ \forall s \in S\}$ has a deterministic decomposed policy as optimum solution.*

□

In other words, if in any state $s \in S$, one finds a set of constraints $\{\mathring{B}_s \mathring{p}_s \leq \mathring{b}_s\}$ defining a $\{0,1\}$-polytope constraining in average the feasible policies p to be in the reduced action space $A'(s)$, then from the state policy decomposition criteria of Corollary 5, solving the decomposed dual LP (A.8) together with constraints $\{\mathring{B}_s \mathring{\pi}_s \leq \mathring{b}_s \mathring{\varpi}_s,\ \forall s \in S\}$ will provide an optimal deterministic policy in A'. We use this sufficient condition to prove Theorem 12.

Proof of Theorem 12: $\mathring{p}_s^i(a_i) = \frac{\mathring{\pi}_s^i(a_i)}{\mathring{\varpi}_s}$ is the discrete probability to take action a_i out of all actions $A_i(s)$ in a state s. Therefore, in state s, for an action reduction constraint (s, R, m, M), Equation (A.11) reduces the solution space to the decomposed randomized policies that select in average at least m and at most M actions out of the set R: $m \leq \sum_{a_i \in R} \mathring{p}_s^i(a_i) \leq M$.

For each state $s \in S$, we rewrite the polytope $\{m \leq \sum_{a_i \in R_j} \mathring{p}_s^i(a_i) \leq M,\ j \in J\}$ in the canonical form $\{\mathring{B}_s \mathring{p}_s \leq \mathring{b}_s\}$. We use the total unimodularity theory (Schrijver, 2003). If no sub-action is present in more than one action reduction constraint, the $\{-1,0,1\}$-matrix $\mathring{B}_s$ has in each column either exactly one 1 and one -1 or only 0 values. $\mathring{B}_s$ can then be seen as the incidence matrix of an oriented graph that is totally unimodular. Since vector $\mathring{b}_s$ is integral, $\{\mathring{B}_s \mathring{p}_s \leq \mathring{b}_s\}$ defines a polyhedron with $\{0,1\}$-vector vertices. Applying Corollary 5, deterministic policies are then dominant. □

A.4.3 Complexity and efficiency of action space reduction

Theorem 13 states that there exists a set of constraints to add in the decomposed dual LP (A.8) such that it will return the best policies in A' and that this policy will be deterministic. However, we prove now that in general the polyhedral description of a subset of decomposed policies can be less efficient than the non-decomposed ones.

Proposition 11. *The number of constraints necessary to describe the convex hull of a subset of decomposed policies can be greater than the number of corresponding non-decomposed policies.*

Proof. There is a positive constant c such that there exist $\{0,1\}$-polytopes in dimension n with $(\frac{cn}{\log n})^{\frac{n}{4}}$ facets (Bárány and Pór, 2001), while the number of vertices is less than 2^n. □

In practice, we saw in our dynamic pricing example that one can formulate valid inequalities. One can use Corollary 5 to check if the decomposed dual LP (A.8) together with for instance Equations (A.12b) correctly models the action space reduction. However, applying Corollary 5 implies to check if these constraints define a polyhedron with $\{0,1\}$-vertices. We investigate now the complexity of checking this sufficient condition. From Papadimitriou and Yannakakis (1990), we know that determining whether a polyhedron $\{x \in \mathbb{R}^n : \; Ax \leq b\}$ is integral is co-NP-complete. In the next lemma we show that it is also co-NP-complete for $\{0,1\}$-polytopes as a direct consequence of Ding *et al.* (2008).

Lemma 13. *Determining whether a polyhedron* $\{x \in \mathbb{R}^n : \; Ax \leq b, \; 0 \leq x \leq 1\}$ *is integral is co-NP-complete.*

Proof. Let A' be a $\{0,1\}$-matrix with precisely two ones in each column. From Ding *et al.* (2008) we know that the problem of deciding whether the polyhedron $P = \{x : \; A'x \geq 1, \; x \geq 0\}$ is integral is co-NP-complete. Note that all vertices v of P respect $0 \leq v \leq 1$. Therefore, $\{x : \; A'x \geq 1, \; x \geq 0\}$ is integral if and only if $\{x : \; A'x \geq 1, \; 0 \leq x \leq 1\}$ is integral. It means that determining whether the polyedron P defined by the linear system $\{x : \; A'x \geq 1, \; 0 \leq x \leq 1\}$ is integral is co-NP-complete. The latter problem is a particular case of determining whether for a general matrix A a polyhedron $\{x : \; Ax \leq b, \; 0 \leq x \leq 1\}$ is integral. □

We now use this lemma to establish the complexity of checking the condition of Corollary 5.

Proposition 12. *Let B be a matrix, b be a vector , $\mathring{p} \in \mathbb{R}^n$ and $\{A_i \mid i \in I\}$ be a $|I|$-partition of the n coordinates of vector $\mathring{p}$, i.e. $\mathring{p} = \Big(\mathring{p}(a_i), i \in I, a_i \in A_i\Big)$. Deciding whether polyhedron*

$$\left\{\mathring{p} \in \mathbb{R}^n : \; B\mathring{p} \leq b, \; \sum_{a_i \in A_i} \mathring{p}(a_i) = 1, \; \forall i \in I, \; \mathring{p} \geq 0\right\}$$

has only $\{0,1\}$-vertices is co-NP-complete.

Proof. We reduce the co-NP-complete problem (Lemma 13) of determining whether a polyhedron $\{x \in \mathbb{R}^n : \; Ax \leq b, \; 0 \leq x \leq 1\}$ has only $\{0,1\}$-vertices to the problem of determining whether polyhedron $\left\{x \in \mathbb{R}^{n+1} : \; A'x \leq b', \; \sum_{i=1}^{n+1} x_i = 1, \; x \geq 0\right\}$ has $\{0,1\}$-vertices. The linear system $A'x \leq b'$ has the same equations as $Ax \leq b$ plus $x_{n+1} = 1 - \sum_{i=1}^{n} x_i$. This is a particular case where $|I| = 1$ of deciding whether polyhedron $\left\{\mathring{p} \in \mathbb{R}^n : \; B\mathring{p} \leq b, \; \sum_{a_i \in A_i} \mathring{p}(a_i) = 1, \; \forall i \in I, \; \mathring{p} \geq 0\right\}$ has $\{0,1\}$-vertices, that is hence also co-NP-complete. □

To use the sufficient condition of Corollary 5, we need to check if the vertices of the polyhedron $\{\mathring{B}_s\mathring{p}_s \leq \mathring{b}_s\}$ are $\{0,1\}$-vectors for each state $s \in S$. From Proposition 12, for each state $s \in S$, it amounts then to solving a co-NP-complete problem.

In fact, it is even NP-complete to determine if this polyhedron contains a deterministic policy solution.

Proposition 13. *Consider a decomposed CTMDP with extra constraints of the form $\{B_s \mathring{p}_s \leq b_s,\ \forall s \in S\}$. Determining if there exists a feasible deterministic policy solution of this decomposed CTMDP is NP-complete even if $|S| = 1$.*

Proof. We show a reduction to the well known NP-complete problem 3-SAT. We reduce a 3-SAT instance with a set V of n variables and m clauses to a D-CTMDP instance. The system is composed with only one state s, so $\varpi_s = 1$. Each variable v creates an independent sub-action set A_v containing two sub-actions representing the two possible states (literal l) of the variable: v and $\bar{v}$. We have then $A = \prod_{v \in V} A_v = \prod_{v \in V} \{v, \bar{v}\}$. Each clause C generates a constraint: $\sum_{l \in C} l \geq 1$. Finally, there exists a deterministic feasible policy for the D-CTMDP instance if and only if the 3-SAT instance is satisfiable. □

A.5 Numerical experiments

In this section we compare the efficiency of the LP formulation and the dynamic formulation with both the classic and decomposed formulation for the multi-class M/M/1 queue dynamic pricing example detailed in the previous sections. We create instances with n classes of clients and with the set of k prices $\mathcal{P} = \{2i,\ i \in \{0, \ldots, k-1\}\}$. Clients of class i with price $r_i \in \mathcal{P}$ arrive according to an independent Poisson process with rate $\lambda^i(r_i) = (4 - i)(10 - r_i)$, except for the price 0 which means that we are refusing a client: *i.e.* $\lambda^i(0) = 0$. For a client of class i the waiting cost per unit of time is $h^i = 2^{4-i}$ and his processing time is exponentially distributed with rate $\mu^i = 20 - 4i$.

Algorithms are tested on an Intel core 2 duo 2.4 Ghz processor with 2 GB of RAM. Heuristics are written in JAVA and the LP is solved with Gurobi 4.6. Legend (F-M) has to be read as follows: F∈{C, D} stands for Formulation, C for Classic or D for Decomposed; M∈{VI-ϵ, LP} stands for Method, VI-ϵ for Value Iteration at precision ϵ and LP for Linear Programming.

We compare the computation time of the different algorithms on the same instances. We confront 6 solution methods: the classic and decomposed value iteration algorithms for two values of ϵ: 10^{-2} and 10^{-5}, and the classic and decomposed dual LP formulation.

First, for both the classic and the decomposed formulation, the value iteration computation time depends on the precision asked: dividing ϵ per 1000 increases roughly the computation time by a factor 2. We also clearly see that the value iteration algorithm is much quicker to solve than the LP formulation.

Secondly benefits of the decomposition appear obvious. When the number of states grows, variations on the queue capacity C (Table A.1) or the number of classes n (Table A.2) influence less the decomposed formulation. It is even clearer when we increase the number of proposed prices k, indeed as shown in Table A.3, the difference of computation time between the classic and the decomposed formulations increases exponentially with k.

Finally, in Table A.4 we study a D-CTMDPs with reduced action space. We take decomposable instance with (C=5, n=4, k=4) and study two action space reductions: the case where we forbid to have all high prices selected in a same state ($|\mathcal{P}'| = |\mathcal{P}| - 1$) and the case where we want to select exactly $n/2$ high prices (($|\mathcal{P}'| \approx |\mathcal{P}|/2$)). Decomposed DP formulation are in this case Non Applicable (NA). Table A.4 reports the important benefit in term of computation time of using the decomposed LP formulation.

(C,n,k)	C-VI-10^{-2}	D-VI-10^{-2}	C-VI-10^{-5}	D-VI-10^{-5}	C-LP	D-LP
(5,3,4)	0.79	0.09	2.16	0.71	4.94	0.27
(10,3,4)	18.02	1.05	34.89	1.70	101.8	7.08
(15,3,4)	82.71	4.24	143.2	7.22	4244	290

Table A.1: Influence of the queue capacity C on the algorithms computation time (in s.).

(C,n,k)	C-VI-10^{-2}	D-VI-10^{-2}	C-VI-10^{-5}	D-VI-10^{-5}	C-LP	D-LP
(10,1,4)	0	0	0	0	0.03	0.03
(10,2,4)	0.07	0.01	0.08	0.02	0.36	0.13
(10,3,4)	18.02	1.05	34.89	1.70	101.8	7.08
(5,4,4)	87.9	0.89	383	3.56	541.7	3.72

Table A.2: Influence of the number of classes n on the algorithms computation time (in s.).

(C,n,k)	C-VI-10^{-2}	D-VI-10^{-2}	C-VI-10^{-5}	D-VI-10^{-5}	C-LP	D-LP
(10,3,2)	2.51	0.45	2.67	0.54	7.1	1.1
(10,3,3)	6.12	0.55	10.60	0.81	20.4	2.6
(10,3,4)	18.02	1.05	34.89	1.70	101.8	7.08
(10,3,5)	37.56	1.2	80.23	2.03	331	9.7

Table A.3: Influence of the number of prices k on the algorithms computation time (in s.).

$\|\mathcal{P}'\|$	C-VI-10^{-2}	D-VI-10^{-2}	C-VI-10^{-5}	D-VI-10^{-5}	C-LP	D-LP
$\|\mathcal{P}\|-1$	81.2	NA	357.6	NA	503.1	3.15
$\|\mathcal{P}\|/2$	45.3	NA	165.5	NA	279.8	3.7

Table A.4: Computation time (in s.) to solve a CTMDP (C=5, n=4, k=4) with a reduced action space $\mathcal{P}'$.

A.6 Discounted reward criterion

We extends our results now to the discounted reward criterion, *i.e.* when future rewards are discounted by factor $\beta \in]0,1[$. In this section, we use a positive scalar α_s, $s \in S$, which satisfies $\sum_{s\in S} \alpha_s = 1$. Any other positive constant would work but when the sum is equal to 1 it allows an interpretation as an initial state probability distribution over the states S.

A.6.1 Classic CTMDP

A.6.1.1 Optimality equations

To write down the optimality equations, in state s we use an unifomization with rate $\Lambda_s := \sum_{t\in S} \Lambda_{s,t}$ with:

$$\Lambda_{s,t} := \max_{a\in A(s)} \lambda_{s,t}(a) = \max_{\substack{(a_1,\ \dots,\ a_n) \\ \in\ A_1(s)\times\ldots\times A_n(s)}} \sum_{i\in I} \lambda^i_{s,t}(a_i) = \sum_{i\in I} \max_{a_i\in A_i(s)} \lambda^i_{s,t}(a_i).$$

We have then that the optimal expected discounted reward per state v^* satisfies the optimality equations:

$$\forall s \in S, \quad v(s) = T\Big(v(s)\Big), \tag{A.14}$$

with the operator T defined $\forall s \in S$ as follows:

$$T\Big(v(s)\Big) = \max_{a \in A(s)} \left\{ \frac{1}{\beta + \Lambda_s} \left(\tilde{h}_s(a) + \sum_{t \in S} \Big[\lambda_{s,t}(a) v(t) + \Big(\Lambda_{s,t} - \lambda_{s,t}(a)\Big) v(s) \Big] \right) \right\}.$$

To compute the best MDP policy we can use the value iteration algorithm on optimality equations (A.14). It is the same scheme as defined for the average reward criterion in Section A.2.2.

A.6.1.2 LP formulation

Under some general conditions, the optimality equations (A.14) have a solution and the optimal expected reward per state v^* is the vector v with the smallest value $\sum_{s \in S} \alpha_s v(s)$ which satisfies:

$$v(s) \geq T\Big(v(s)\Big), \quad \forall s \in S. \tag{A.15}$$

We can linearize the max function of operator T in equations (A.15) to formulate the following LP which has for optimal solution v^*:

Primal LP

$$\begin{aligned} \min \ & \sum_{s \in S} \alpha_s v(s) \\ \text{s.t.} \ & \left(\beta + \sum_{t \in S} \lambda_{s,t}(a) \right) v(s) - \sum_{t \in S} \lambda_{s,t}(a) v(t) \geq \tilde{h}_s(a), \qquad \forall s \in S,\ \forall a \in A(s), \\ & v(s) \in \mathbb{R}. \end{aligned}$$

Dual LP

$$\begin{aligned} \max \ & \sum_{s \in S} \sum_{a \in A(s)} \tilde{h}_s(a) \widetilde{\pi}_s(a) \\ \text{s.t.} \ & \sum_{a \in A(s)} \left(\beta + \sum_{t \in S} \lambda_{s,t}(a) \right) \widetilde{\pi}_s(a) - \sum_{t \in S} \sum_{a \in A(t)} \lambda_{t,s}(a) \widetilde{\pi}_t(a) = \alpha_s, \qquad \forall s \in S, \\ & \widetilde{\pi}_s(a) \geq 0. \end{aligned}$$

We can interpret the dual variables $\widetilde{\pi}_s(a)$ as the total discounted joint probability under initial state distributions α_s that the system occupies state $s \in S$ and chooses action $a \in A(s)$. Some other interpretations can be retrieved in Puterman (1994).

We could have also constructed the previous dual LP with variable substitutions from the following QP:

QP

$$
\begin{aligned}
\max \; & \sum_{s\in S}\sum_{a\in A(s)} \tilde{h}_s(a)\widetilde{\varpi}_s p_s(a) \\
\text{s.t. } & \sum_{a\in A(s)} \left(\beta + \sum_{t\in S}\lambda_{s,t}(a)\right)\widetilde{\varpi}_s p_s(a) - \sum_{t\in S}\sum_{a\in A(t)} \lambda_{t,s}(a)\widetilde{\varpi}_t p_t(a) = \alpha_s, \quad \forall s\in S, \\
& \sum_{s\in S}\sum_{a\in A(s)} p_s(a) = 1, \\
& \widetilde{\varpi}_s \geq 0, \quad p_s(a) \geq 0.
\end{aligned}
$$

A.6.2 Action Decomposed CTMDP

A.6.2.1 Optimality equations

To use the action decomposability we rewrite optimality equations (A.14) with an explicit decomposition. Using the same uniformization as in the classic case we obtain a decomposed operator T: $\forall s \in S$,

$$
\begin{aligned}
T\Big(v(s)\Big) &= \max_{\substack{(a_1,\ \dots,\ a_n) \\ \in\ A_1(s)\times\dots\times A_n(s)}} \left\{ \sum_{i=1}^{n} \left[\frac{1}{\beta+\Lambda_s}\left(\tilde{h}^i_s(a_i) + \sum_{t\in S}\Big[\lambda^i_{s,t}(a_i)v(t) + \Big(\Lambda^i_{s,t} - \lambda^i_{s,t}(a_i)\Big)v(s)\Big]\right)\right]\right\} \\
&= \sum_{i=\in I}\left[\max_{a_i\in A_i(s)} \left\{ \frac{1}{\beta+\Lambda_s}\left(\tilde{h}^i_s(a_i) + \sum_{t\in S}\Big[\lambda^i_{s,t}(a_i)v(t) + \Big(\Lambda^i_{s,t} - \lambda^i_{s,t}(a_i)\Big)v(s)\Big]\right)\right\}\right].
\end{aligned}
$$

To compute the best MDP policy we can now use again the value iteration algorithm but with the decomposed operator that is much more efficient. The optimality equations with the decomposed operator also lead to a LP formulation that we formulate in the next section.

A.6.2.2 LP formulation

Under some general conditions, optimality equations (A.14) with decomposed operator T have a solution and the optimal expected reward per state v^* is the vector v with the smallest value $\sum_{s\in S}\alpha_s v(s)$ which satisfies:

$$
\begin{aligned}
\alpha v^* = \min \quad & \sum_{s\in S}\alpha_s v(s) \\
\text{s.t.} \quad & v(s) \geq T\Big(v(s)\Big), \qquad \forall s \in S.
\end{aligned}
$$

That we can reformulate:

$$\begin{aligned}
&\min \sum_{s\in S}\alpha_s v(s)\\
&\text{s.t.}\quad 0 \geq \max_{s\in S}\left\{T\Big(v(s)\Big) - v(s)\right\}\\
&\qquad \geq \max_{s\in S}\left\{(\beta+\Lambda_s)(T\Big(v(s)\Big) - v(s))\right\}\\
&\qquad \geq \max_{s\in S}\left\{\sum_{i=1}^{n}\left[\max_{a_i\in A_i(s)}\left\{\tilde{h}^i_s(a_i) + \sum_{t\in S}\lambda^i_{s,t}(a_i)\Big(v(t)-v(s)\Big)\right\}\right] - \beta v(s)\right\}.
\end{aligned}$$

Lemma 14. *For any finite sets $S,\ I,\ A$, any data coefficients $\alpha_s,\ \gamma_{s,t,i,a},\ \delta_{s,i,a},\ \zeta_s \in \mathbb{R},\ s,t\in S,\ i\in I$ and $a\in A$, the vector $v\in\mathbb{R}^{|S|}$ with the smallest value $\sum_{s\in S}\alpha_s v(s)$ satisfying*

$$0 \geq \max_{s\in S}\left\{\sum_{i\in I}\max_{a\in A}\left\{\sum_{t\in S}\gamma_{s,t,i,a}v(s) + \delta_{s,i,a}\right\} + \zeta_s v(s)\right\}$$

is the solution of the following LP:

$$\begin{aligned}
&\min \sum_{s\in S}\alpha_s v(s)\\
&s.t.\quad m(s,i) \geq \sum_{t\in S}\gamma_{s,t,i,a}v(s) + \delta_{s,i,a}, && \forall s\in S,\ \forall i\in I,\ \forall a\in A,\\
&\qquad 0 \geq \sum_{i\in I} m(s,i) + \zeta_s v(s), && \forall s\in S,\\
&\qquad m(s,i)\in\mathbb{R},\quad v(s)\in\mathbb{R}.
\end{aligned}$$

Proof. It is clear that we are minimizing $\sum_{s\in S}\alpha_s v(s)$. Now for any vector v we can see that $\forall s\in S,\ \forall i\in I,\ m(s,i) \geq \max_{a\in A}\left\{\sum_{t\in S}\gamma_{s,t,i,a}v(s)+\delta_{s,i,a}\right\}$, and we have $\forall s\in S, \sum_{i\in I} m(s,i) + \zeta_s v(s) \leq 0$. Therefore any vector v solution must satisfy $\forall s\in S\ 0 \geq \sum_{i\in I}\max_{a\in A}\left\{\sum_{t\in S}\gamma_{s,t,i,a}v(s)+\delta_{s,i,a}\right\} + \zeta_s v(s)$ and finally $0 \geq \max_{s\in S}\left\{\sum_{i\in I}\max_{a\in A}\left\{\sum_{t\in S}\gamma_{s,t,i,a}v(s)+\delta_{s,i,a}\right\} + \zeta_s v(s)\right\}$. ☐

Using Lemma 14 we obtain that v^* is the solution of the following LP.

Decomposed Primal LP

$$\begin{aligned}
&\min \sum_{s\in S}\alpha_s v(s)\\
&\text{s.t. } m(s,i) \geq \tilde{h}^i_s(a_i) + \sum_{t\in S}\lambda^i_{s,t}(a_i)\Big(v(t)-v(s)\Big), && \forall s\in S,\ \forall i\in I,\ \forall a_i\in A_i(s),\\
&\beta v(s) - \sum_{i\in I} m(s,i) \geq 0, && \forall s\in S \leq 0,\\
&m(s,i)\in\mathbb{R},\quad v(s)\in\mathbb{R}.
\end{aligned}$$

Decomposed Dual LP

$$\max \sum_{s\in S}\sum_{i\in I}\sum_{a_i\in A_i(s)} \mathring{\tilde{\pi}}^i_s(a_i)\tilde{h}^i_s(a_i)$$

$$\text{s.t.} \sum_{a_i\in A_i(s)} \mathring{\tilde{\pi}}^i_s(a_i) = \mathring{\tilde{\varpi}}_s, \qquad \forall s\in S,\ \forall i\in I,$$

$$\beta\mathring{\tilde{\varpi}}_s + \sum_{i\in I}\sum_{a_i\in A_i(s)}\sum_{t\in S}\lambda^i_{s,t}(a_i)\mathring{\tilde{\pi}}^i_s(a_i) - \sum_{t\in S}\sum_{i\in I}\sum_{a_i\in A_i(t)}\lambda^i_{t,s}(a_i)\mathring{\tilde{\pi}}^i_t(a_i) = \alpha_s, \qquad \forall s\in S,$$

$$\mathring{\tilde{\pi}}^i_s(a_i)\geq 0, \quad \mathring{\tilde{\varpi}}_s \geq 0.$$

Under initial state distributions α, we can interpret $\mathring{\tilde{\varpi}}_s$ as the total discounted joint probability that the system occupies state $s\in S$ under initial state distributions α_s, and $\mathring{\tilde{\pi}}^i_s(a_i)$ as the total discounted joint probability that the system occupies state $s\in S$ and chooses action $a_i\in A_i(s)$. The decomposed dual LP has $|S|(kn+1)$ variables and $|S|((k+1)n+2)$ constraints. It is much less than the classic dual LP that has $|S|k^n$ variables and $|S|(k^n+1)$ constraints.

We could have also constructed the previous dual decomposed LP with variable substitutions from the following QP:

Decomposed QP

$$\max \sum_{s\in S}\sum_{i\in I}\sum_{a_i\in A_i(s)} \tilde{h}^i_s(a_i)\mathring{p}^i_s(a_i)\mathring{\tilde{\varpi}}_s$$

$$\text{s.t.}\ \beta\mathring{\tilde{\varpi}}_s + \sum_{i\in I}\sum_{a_i\in A_i(s)}\sum_{t\in S}\lambda^i_{s,t}(a_i)\mathring{p}^i_s(a_i)\mathring{\tilde{\varpi}}_s - \sum_{t\in S}\sum_{i\in I}\sum_{a_i\in A_i(t)}\lambda^i_{t,s}(a_i)\mathring{p}^i_t(a_i)\mathring{\tilde{\varpi}}_t = \alpha_s, \qquad \forall s\in S,$$

$$\sum_{a_i\in A_i(s)}\mathring{p}^i_s(a_i) = 1, \qquad \forall s\in S,\ \forall i\in I,$$

$$\mathring{p}^i_s(a_i)\geq 0, \quad \mathring{\tilde{\varpi}}_s\geq 0.$$

Remark 10. *Theorems 11, 13, 12, Corollary 5 and Propositions 11, 12, 13, are also applicable when considering the discounted reward criterion with the substitution* $(\widetilde{\pi},\widetilde{\varpi})\to(\pi,\varpi)$, $(\mathring{\tilde{\pi}},\mathring{\tilde{\varpi}})\to(\mathring{\pi},\mathring{\varpi})$ *and hence with* $\mathring{p}_s(a) = \frac{\mathring{\tilde{\pi}}_s(a)}{\mathring{\tilde{\varpi}}_s}$ *and* $\mathring{p}^i_s(a_i) = \frac{\mathring{\tilde{\pi}}^i_s(a_i)}{\mathring{\tilde{\varpi}}_s}$.

Bibliography

Anderson, E., Nash, P., and Perold, A. (1983). Some properties of a class of continuous linear programs. *SIAM Journal on Control and Optimization*, **21**(5), 758–765. (Cited on page 112.)

Anderson, E. J. and Nash, P. (1987). *Linear programming in infinite-dimensional spaces: theory and applications*. John Wiley and Sons, New York, NY. (Cited on page 112.)

Autolib' (2011). http://www.autolib-paris.fr. (Cited on pages 2, 8, 15, 17, 19, 20, 21, and 27.)

Balcan, M.-F. and Blum, A. (2006). Approximation algorithms and online mechanisms for item pricing. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pages 29–35. ACM. (Cited on page 71.)

Balsamo, S., de Nitto Personé, V., and Onvural, R. (2000). *Analysis of queueing networks with blocking*, volume 31. Springer. (Cited on page 46.)

Bampou, D. and Kuhn, D. (2012). Polynomial approximations for continuous linear programs. *SIAM Journal on Optimization*, **22**(2), 628–648. (Cited on page 112.)

Bárány, I. and Pór, A. (2001). On 0-1 polytopes with many facets. *Advances in Mathematics*, **161**(2), 209–228. (Cited on page 180.)

Barclays Cycle Hire (2010). http://www.tfl.gov.uk/roadusers/cycling/14808.aspx. (Cited on page 124.)

Baskett, F., Chandy, K., Muntz, R., and Palacios-Gomez, F. (1975). Open, closed, and mixed networks of queues with different classes of customers. *Journal of the Association for Computing Machinery*, **22**. (Cited on pages 44, 50, and 88.)

Bäuerle, N. (2000). Asymptotic optimality of tracking policies in stochastic networks. *The Annals of Applied Probability*. (Cited on page 105.)

Bäuerle, N. (2002). Optimal control of queueing networks: An approach via fluid models. *Advances in Applied Probability*. (Cited on page 105.)

Bellman, R. (1953). Bottleneck problems and dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, **39**(9), 947. (Cited on pages 51, 112, and 159.)

Bertsekas, D. P. (2005a). *Dynamic programming and optimal control. Vol. I.* 3rd ed., Athena Scientific, Belmont, MA. (Cited on page 45.)

Bertsekas, D. P. (2005b). *Dynamic programming and optimal control. Vol. II.* 3rd ed., Athena Scientific, Belmont, MA. (Cited on pages 161, 164, 165, and 172.)

Bertsekas, D. P. and Castañon, D. A. (1989). Adaptive aggregation methods for infinite horizon dynamic programming. *Automatic Control, IEEE Transactions on*, **34**(6), 589–598. (Cited on page 163.)

Bertsimas, D., Gamarnik, D., and Rikun, A. A. (2011a). Performance analysis of queueing networks via robust optimization. *Operations Research*, **59**(2), 455–466. (Cited on page 61.)

Bertsimas, D., Brown, D. B., and Caramanis, C. (2011b). Theory and applications of robust optimization. *SIAM Rev.*, **53**(3), 464–501. (Cited on page 61.)

Bixi (2009). http://montreal.bixi.com. (Cited on pages 21, 25, 29, and 61.)

Budget Truck Rental (1998). http://www.budgettruck.com. (Cited on page 29.)

Büyüktahtakin, I. E. (2011). *Dynamic Programming Via Linear Programming*. John Wiley & Sons, Inc. (Cited on page 175.)

Buzen, J. (1973). Computational algorithms for closed queueing networks with exponential servers. *Communications of the ACM 16*, page 527–531. (Cited on page 51.)

Capital Bikeshare (2010). http://www.capitalbikeshare.com. (Cited on pages 125 and 126.)

Car2go (2008). http://www.car2go.com. (Cited on pages 2, 8, 15, 17, 18, 20, and 21.)

Carroll, W. and Grimes, R. (1995). Evolutionary change in product management: Experiences in the car rental industry. *Interfaces*, **25**(5), 84–104. (Cited on page 29.)

Çil, E. B., Karaesmen, F., and Örmeci, E. L. (2011). Dynamic pricing and scheduling in a multi-class single-server queueing system. *Queueing Systems - Theory and Applications*, **67**, 305–331. (Cited on page 161.)

Chemla, D. (2012). *Algorithms for optimizing shared mobility systems*. Ph.D. thesis, Université Paris-Est, France. (Cited on page 25.)

Chemla, D., Meunier, F., and Wolfler Calvo, R. (2012). Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization*. (Cited on pages 2, 8, and 25.)

Chemla, D., Meunier, F., Pradeau, T., Wolfer Calvo, R., and Yahiaoui, H. (2013). Self-service bike sharing systems: Simulation, repositioning, pricing. (Cited on pages 3, 9, 31, 123, 147, and 154.)

Cité lib (2010). http://citelib.com. (Cited on page 16.)

Citybike Wien (2003). http://www.citybikewien.at. (Cited on pages 21 and 124.)

Côme, E. (2012). Model-based clustering for BSS usage mining: a case study with the vélib' system of paris. In *International workshop on spatio-temporal data mining for a better understanding of people mobility: The Bicycle Sharing System (BSS) case study. Dec 2012*. (Cited on pages 2, 8, 22, 23, 127, 128, 129, 149, and 156.)

Contardo, C., Morency, C., and Rousseau, L.-M. (2012). Balancing a dynamic public bike-sharing system. Technical Report 09, CIRRELT. (Cited on pages 2, 8, and 25.)

CRC, C. (2012). Rapport sur la gestion de Vélib'. Technical report, Ville de Paris. (Cited on pages 27 and 28.)

DeMaio, P. (2009). Bike-sharing: History, impacts, models of provision, and future. *Journal of Public Transportation*, **12**(4), 41–56. (Cited on pages 1, 2, 7, and 16.)

Ding, G., Feng, L., and Zang, W. (2008). The complexity of recognizing linear systems with certain integrality properties. *Mathematical Programming, Series A*, **114**, 321–334. (Cited on page 181.)

Dos Santos Eleutério, V. L. (2009). *Finding Approximate Solutions for Large Scale Linear Programs*. Ph.D. thesis, ETH Zurich. (Cited on pages 162 and 175.)

Edmonds, J. and Karp, R. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (Association for Computing Machinery)*, **19**(2), 248–264. (Cited on page 90.)

Efthymiou, D., Antoniou, C., and Tyrinopoulos, Y. (2012). Spatially aware model for optimal site selection. *Transportation Research Record: Journal of the Transportation Research Board*, **2276**(1), 146–155. (Cited on page 24.)

Filippi, C. (2011). *Sensitivity Analysis in Linear Programming*. Wiley Encyclopedia of Operations Research and Management Science. (Cited on pages 162 and 175.)

Fishman, E., Washington, S., and Haworth, N. (2012). Barriers and facilitators to public bicycle scheme use: A qualitative approach. *Transportation Research Part F: Traffic Psychology and Behaviour*, **15**(6), 686 – 698. (Cited on page 126.)

Fleischer, L. and Sethurama, J. (2005). Efficient algorithms for separated continuous linear programs: the multicommodity flow problem with holding costs and extensions. *Mathematics of Operations Research*, **30:4**, 916–938. (Cited on pages 112 and 119.)

Fricker, C. and Gast, N. (2012). Incentives and regulations in bike-sharing systems with stations of finite capacity. *arXiv :1201.1178v1*. (Cited on pages 2, 3, 8, 9, 24, 32, 44, 45, 46, 130, 147, 154, and 155.)

Fricker, C., Gast, N., and Mohamed, H. (2012). Mean field analysis for inhomogeneous bike sharing systems. In *23rd Intern. Meeting on Probabilistic, Combinatorial, and Asymptotic Methods for the Analysis of Algorithms (AofA'12)*. (Cited on pages 24, 45, 124, 131, 147, and 155.)

Gallego, G. and van Ryzin, G. (1994). Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, **40**(8), 999–1020. (Cited on pages 105 and 118.)

Garey, M. R. and Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. A Series of books in the mathematical sciences. W. H. Freeman, San Francisco. (Cited on page 66.)

George, D. K. and Xia, C. H. (2011). Fleet-sizing and service availability for a vehicle rental system via closed queueing networks. *European Journal of Operational Research*, **211**(1), 198 – 207. (Cited on pages 2, 8, 24, 44, 50, 55, 85, 88, 90, 96, 97, and 131.)

Geraghty, M. K. and Johnson, E. (1997). Revenue management saves national car rental. *Interfaces*, **27**(1997 vol. 27 no. 1), 107–127. (Cited on page 29.)

Glebov, N. (1973). About one class of convex integer programs. *Upravlyaemye Sistemy, Vol. 11, in: Institute of Mathematics of Siberian Branch of Academy of Sciences of the USSR, Novosibirsk*, pages 38–42 (in Russian). (Cited on page 95.)

Green, L. and Kolesar, P. (1991). The pointwise stationary approximation for queues with nonstationary arrivals. *Management Science*, **37**(1), 84–97. (Cited on pages 46 and 116.)

Guerriero, F., Miglionico, G., and Olivito, F. (2012). Revenue management policies for the truck rental industry. *Transportation Research Part E: Logistics and Transportation Review*, **48**(1), 202 – 214. (Cited on page 29.)

Guo, X. and Hernández-Lerma, O. (2009). *Continuous-Time Markov Decision Processes: Theory and Applications*. Stochastic modelling and applied probability. Springer Verlag. (Cited on pages 161 and 164.)

Guruswami, V., Hartline, J., Karlin, A., Kempe, D., Kenyon, C., and McSherry, F. (2005). On profit-maximizing envy-free pricing. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1164–1173. Society for Industrial and Applied Mathematics. (Cited on page 71.)

Haensela, A., Mederer, M., and Schmidt, H. (2011). Revenue management in the car rental industry: A stochastic programming approach. (Cited on page 29.)

Hangzhou Public Bicycle (2008). http://www.hzzxc.com.cn. (Cited on pages 16, 17, and 19.)

Hertz (1918). http://www.hertz.com. (Cited on page 16.)

Hertz on Demand (2008). http://www.hertzondemand.com. (Cited on page 16.)

Hu, J., Fu, M. C., Ramezani, V. R., and Marcus, S. I. (2007). An evolutionary random policy search algorithm for solving markov decision processes. *INFORMS Journal on Computing*, **19**(2), 161–174. (Cited on page 163.)

Ion, L., Cucu, T., Boussier, J.-M., Teng, F., Breuil, D., *et al.* (2009). Site selection for electric cars of a car-sharing service. In *Proceedings of EVS24: International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium & Exhibition*. (Cited on page 24.)

Jacqué, P. (2013). Avec bluecar, vincent bolloré veut avant tout populariser sa technologie de batterie. *Le Monde*. (Cited on page 27.)

Karloff, H. and Zwick, U. (1997). A 7/8-approximation algorithm for MAX 3SAT? In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science*, pages 406 –415. (Cited on page 68.)

Kaspi, M., Raviv, T., and Tzur, M. (2013). Vehicle sharing systems - reservations are good! *Manuscript submitted for publication*. (Cited on page 25.)

Khandekar, R., Kimbrel, T., Makarychev, K., and Sviridenko, M. (2009). On hardness of pricing items for single-minded bidders. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 202–216. (Cited on pages 71 and 72.)

Koole, G. M. (1998). Structural results for the control of queueing systems using event-based dynamic programming. *Queueing Systems*. (Cited on pages 51, 159, 161, 168, and 170.)

Kumar, V. P. and Bierlaire, M. (2012). Optimizing locations for a vehicle sharing system. In *Swiss Transport Research Conference*. (Cited on pages 2, 8, and 24.)

Li, C. and Neely, M. J. (2012). Delay and rate-optimal control in a multi-class priority queue with adjustable service rates. In A. G. Greenberg and K. Sohraby, editors, *INFOCOM*, pages 2976–2980. IEEE. (Cited on page 161.)

Lin, J.-R. and Yang, T.-H. (2011). Strategic design of public bicycle sharing systems with service level constraints. *Transportation research part E: logistics and transportation review*, **47**(2), 284–294. (Cited on pages 2, 8, and 24.)

Lippman, S. A. (1975). Applying a new device in the optimization of exponential queuing systems. *Operation Research*. (Cited on pages 161, 162, and 165.)

Liu, Y. (2011). *Many-Server Queues with Time-Varying Arrivals, Customer Abandonment and Non-Exponential Distributions*. Ph.D. thesis, COLUMBIA UNIVERSITY, USA. (Cited on page 46.)

Luo, X. and Bertsimas, D. (1999). A new algorithm for state-constrained separated continuous linear programs. *S/AM Journal on control and optimization*, pages 177–210. (Cited on page 112.)

Maglaras, C. (2006). Revenue management for a multiclass single-server queue via a fluid model analysis. *Operations Research*. (Cited on pages 62, 105, 117, and 161.)

Maglaras, C. and Meissner, J. (2006). Dynamic pricing strategies for multiproduct revenue management problems. *Manufacturing & Service Operations Management*, **8**(2), 136–148. (Cited on page 117.)

Manne, A. S. (1960). Linear programming and sequential decisions. *Operations Research*. (Cited on page 166.)

McMahon, J. J. (2008). *Time-Dependence in Markovian Decision Processes*. Ph.D. thesis, The University of Adelaide, Australia. (Cited on page 46.)

Meyn, S. (1997). *Stability and optimization of queueing networks and their fluid models*, chapter Mathematics of Stochastic Manufacturing Systems Lectures in Applied. American Mathematical Society, Providence Mathem. (Cited on page 105.)

Midgley, P. (2011). Bicycle-sharing schemes: enhancing sustainable mobility in urban areas. Technical report, United Nations, Department of Economic and Social Affairs. (Cited on page 27.)

Morency, C., Trépanier, M., and Godefroy, F. (2011). Insight into the Montreal bikesharing system. In *TRB-Transportation Research Board Annual Meeting, Washington, USA, Paper #11-1238, 17 pages*. (Cited on pages 21, 29, and 61.)

Nair, R. (2010). *Design and analysis of vehicle sharing programs: A system approach*. Ph.D. thesis, Department of Civil & Environmental Engineering, University of Maryland, USA. (Cited on page 25.)

Nair, R. and Miller-Hooks, E. (2011). Fleet management for vehicle sharing operations. *Transportation Science*, **45**(4), 524–540. (Cited on pages 2, 8, and 25.)

Osorio, C. and Bierlaire, M. (2009). An analytic finite capacity queueing network model capturing the propagation of congestion and blocking. *European Journal of Operational Research*, **196**(3), 996 – 1007. (Cited on page 46.)

Osorio, C. and Bierlaire, M. (2010). A simulation-based optimization framework for urban traffic control. *To appear in Operations Research*. (Cited on pages 146 and 153.)

Papadimitriou, C. H. and Yannakakis, M. (1990). On recognizing integer polyhedra. *Combinatorica*, **10**(1), 107–109. (Cited on page 181.)

Papanikolaou, D. (2011). *The Market Economy of Trips*. Master's thesis, Massachusetts Institute of Technology. (Cited on pages 31, 33, and 69.)

Papier, F. and Thonemann, U. W. (2010). Capacity rationing in stochastic rental systems with advance demand information. *Operation Research*, **58**(2), 274–288. (Cited on page 25.)

Pfrommer, J., Warrington, J., Schildbach, G., and Morari, M. (2013). Dynamic vehicle redistribution and online price incentives in shared mobility systems. *arXiv preprint arXiv:1304.3949*. (Cited on pages 3, 9, 25, 32, 124, and 125.)

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, New York, NY. (Cited on pages 45, 51, 55, 87, 92, 161, 162, and 185.)

Raviv, T. (2012). The battery switching station scheduling problem. *Operations Research Letters*, **40**(6), 546 – 550. (Cited on page 18.)

Raviv, T. and Kolka, O. (2013). Optimal inventory management of a bike-sharing station. *IIE Transactions*, (just-accepted). (Cited on page 25.)

Raviv, T., Tzur, M., and Forma, I. A. (2013). Static repositioning in a bike-sharing system: models and solution approaches. *EURO Journal on Transportation and Logistics*, **2**(3), 187–229. (Cited on pages 2, 8, and 25.)

Rentn'Drop (2006). http://www.rentanddrop.com. (Cited on page 29.)

Rudloff, C., Lackner, B., Prandtstetter, M., and Straub, M. (2013). Demand modelling as a basis for optimising the rearrangement of bikes in the vienna bicycle sharing system. In *EURO XXVI, Rome, Italy*. (Cited on pages 124, 149, and 156.)

Schrijver, A. (2003). *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Verlag. (Cited on page 180.)

Shenmaier, V. (2003). A greedy algorithm for some classes of integer programs. *Discrete applied mathematics*, **133**(1), 93–101. (Cited on page 95.)

Shoup, D. (2005). *The High Cost of Free-Parking*. Planners Press, Chicago. (Cited on pages 1 and 7.)

Shu, J., Chou, M., Liu, O., Teo, C., and Wang, I.-L. (2010). Bicycle-sharing system: Deployment, utilization and the value of re-distribution. unpublished paper. (Cited on pages 2, 8, and 24.)

Tsitsiklis, J. N. and Van Roy, B. (1996). Feature-based methods for large scale dynamic programming. *Machine Learning*, **22**(1), 59–94. (Cited on page 163.)

Vélib' (2007). http://www.velib.paris.fr. (Cited on pages 1, 2, 7, 8, 16, 17, 22, 25, 27, 28, 32, 127, 128, and 132.)

Vogel, P., Greiser, T., and Mattfeld, D. C. (2011). Understanding bike-sharing systems using data mining: Exploring activity patterns. *Procedia-Social and Behavioral Sciences*, **20**, 514–523. (Cited on page 21.)

Wang, A.-L. (2001). How much can be taught about stochastic processes and to whom. *Training researchers in the use of statistics*, pages 73–85. (Cited on page 37.)

Waserhole, A. and Jost, V. (2013a). Pricing in vehicle sharing systems: Optimization in queuing networks with product forms. (Cited on page 84.)

Waserhole, A. and Jost, V. (2013b). Vehicle sharing system pricing regulation: A fluid approximation. (Cited on pages 62, 101, 105, and 123.)

Waserhole, A., Gayon, J. P., and Jost, V. (2013a). Linear programming formulations for queueing control problems with action decomposability. (Cited on page 160.)

Waserhole, A., Jost, V., and Brauner, N. (2013b). Vehicle sharing system optimization: Scenario-based approach. (Cited on page 61.)

Weber, R. and Stidham, S. (1987). Optimal control of service rates in networks of queues. *Advanced applied probabilities*, **19**, 202–218. (Cited on page 161.)

Weiss, G. (2008). A simplex based algorithm to solve separated continuous linear programs. *Mathematical Programming*, **115**(1), 151–198. (Cited on page 112.)

Yoon, S. and Lewis, M. E. (2004). Optimal pricing and admission control in a queueing system with periodically varying parameters. *Queueing Systems*, **47**(3), 177–199. (Cited on page 46.)

Zipcar (2000). http://www.zipcar.com. (Cited on page 16.)